

Doppler Ultrasound Evaluation of the Haemodynamics of Fainting

Comparing methods of image processing of Doppler ultrasound waveforms using Matlab, to extract a maximum velocity envelope.

Raymond Pointon

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science
in
Physics
at the
University of Canterbury,
Christchurch, New Zealand
November 2004

CONTENTS

CONTENTS.....	ii
ABSTRACT.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
GLOSSARY.....	xii
CHAPTER ONE Background.....	1
1.1 Vasovagal syncope.....	1
1.2 Doppler ultrasound.....	2
1.3 Indexes	3
CHAPTER TWO Threshold Methods.....	5
2.1 Preparing images.....	5
2.2 Ten-pixel-average method.....	7
2.3 Edge filter method.....	10
2.4 Differential method.....	13
2.5 Percentile method.....	15
CHAPTER THREE Curve fitting and ratios.....	19
3.1 Least squares.....	19
3.2 Areas and ratios.....	20
3.3 Loupas index.....	22
3.4 Average pixel value.....	23
3.4.1 Noisy images.....	23
3.4.2 Images with little noise.....	24
3.4.3 Examining more than one cycle in an image.....	25
CHAPTER FOUR Errors and comparisons.....	29
4.1 Clicking errors.....	29
4.2 Comparing the four programs on one image.....	34
4.2.1 Ten-pixel-average method.....	34
4.2.2 Edge detection method.....	37
4.2.3 Differential method.....	41
4.2.4 Percentile method.....	44
4.2.5 Comparison summary.....	47
4.3 Comparing the four programs on a noisy dim image.....	48
4.3.1 Ten-pixel-average method.....	48
4.3.2 Edge detection method.....	50
4.3.3 Differential method.....	51
4.3.4 Percentile method.....	52
4.3.5 Comparison summary.....	53
4.4 A complete image set compared.....	53

CHAPTER FIVE Summary and further work.....	55
5.1 Maximum velocity envelope.....	55
5.2 Area ratios and indexes.....	55
5.3 Area ratio, heart rate, blood pressure and nerve activity.....	56
5.4 Conclusion.....	62
REFERENCES.....	63
APPENDIX ONE A complete image set.....	65
APPENDIX TWO Matlab programs.....	69

ABSTRACT

Every time we stand up, gravity pulls nearly a quarter of the bodies blood supply into the lower body. The heart and circulation must respond within seconds to keep enough blood flowing up to the brain. To meet the challenge the heart will speed up to 10 – 15 beats a minute and the nervous system cause the arteries to narrow so that blood pressure rises. If there is an abnormal response due to a temporary malfunction of the nerves and arteries, blood pressure may drop and fainting can briefly occur.

Tilt testing is often performed to try to reproduce an episode of fainting in patients with this abnormal response. Blood pressure, heart rate, nerve activity and measures of the Superior Mesenteric Artery (SMA) Doppler waveform can be monitored with the subject lying horizontal, and tilted to 60° to try to trigger an episode of loss of consciousness.

The Doppler arterial waveform images can be analysed to provide information about blood flow in or out of the gut (a large vascular bed) and related, along with the other physiological measurements, to the fainting process. A simple ratio, such as end diastolic velocity to peak systolic velocity could be used to measure changes of SMA blood flow. However this simple measure may not be sufficiently discriminatory and a more sophisticated measure, such as the ratio of the area under the curve in systole versus that in diastole will hopefully describe more completely the changes taking place in diastole and hence give a measure of vasoconstriction or vasodilation of the SMA.

In order to do this the Doppler waveforms obtained over time from the SMA need to be processed off-line after being saved as TIFF files for example. Then image processing on the waveform as it appears on the TIFF file, using Matlab, extracts the maximum frequency envelope. From this envelope the cut-off between systole and diastole is determined, the areas of interest calculated and hence the above ratio.

The challenge is to extract a maximum velocity envelope for a range of images, some ‘noisy’, by interactively selecting the better cycles in any given image rather than completely automating the process. The purpose of the research reported here is to create or modify image enhancement techniques to allow a better evaluation of the Doppler waveforms. Three or four methods of producing a maximum velocity envelope will be compared and one chosen on the basis of its speed and ability to accurately process noisy images. The area ratio calculated will also be investigated over a number of images to investigate its sensitivity and measure of blood flow changes.

ACKNOWLEDGEMENTS

I am especially indebted to Dr Lou Reinisch and Dr Isabel Wright for their valuable advice and encouragement throughout the course of this work.

Lou's subtle insight and suggestions have kept me on track throughout the year and provided the inspiration to explore various avenues in the project.

Isabel's cheerful attitude, thorough understanding of Doppler sonography and experience with neural network analysis of Doppler shift signals has also kept this project on track and pointing in the right direction.

My thanks to Dr Richard Watts for his suggestions regarding Fourier analysis and to Dr Nick Cook for discussions about imaging, and spending the time with me to use ultrasound equipment to monitor fluid flow in a Doppler ultrasound phantom.

I also thank Dr Richard Tremewan for suggesting this project for an MSc. thesis and especially thank my wife Mary for her support, and putting up with me over the two years to complete papers and this thesis for an MSc.

LIST OF TABLES

Table 3.1	Five area ratios, average value and standard deviation for one Doppler image...	21
Table 3.2	Average pixel values and area ratios for ten thresholds between 10 and 100 using the ten-pixel-average method on a noisy image.....	24
Table 3.3	Average pixel values and area ratios for ten thresholds between 10 and 100 using the ten-pixel-average method for an image with little noise.....	25
Table 3.4	Mean area ratios and standard deviation for 5 cycles in an image as threshold Changes.....	26
Table 4.2	Twenty trials of clicking in three places on each of the four cycles in an image.	30
Table 4.3	Mean and range of the area ratios for twenty trials in four cycles and the three brightest cycles.....	31
Table 4.4	Twenty trials for clicking twice on the four cycles in an image.....	32
Table 4.5	Mean for the clicking trials using two clicks. Four cycles and three cycles.....	32
Table 4.6	Average pixel and area ratio versus threshold data using the ten-pixel-average method on one cycle in an image.....	36
Table 4.7	Mean area ratios for the four cycles in an image using the ten-pixel-average method.....	37
Table 4.8	Area ratio and threshold data for the edge method used on one cycle in an image.....	40
Table 4.9	Mean area ratio and threshold data for four cycles using the edge method on an image.....	41
Table 4.10	Area ratio and threshold data one cycle in an image using the differential method.....	43
Table 4.11	Average area ratios for all four cycles in an image using the differential method.....	44
Table 4.12	Area ratio, average pixel value and threshold data using the percentile method on one cycle of an image.....	46
Table 4.13	Average area ratio for each threshold for the four cycles in an image using the percentile method.....	47
Table 4.14	Area ratio data for the four programs using the best threshold for that image....	48
Table 4.15	Comparison of area ratios for person 6 using the four methods.....	54
Table 5.1	Area ratio, heart rate, blood pressure and nerve activity for person 15.....	57
Table 5.2	Area ratio, heart rate, blood pressure and nerve activity for person 22.....	60

LIST OF FIGURES

Fig 1.1	Tilt table.....	1
Fig 1.2	Mesenteric artery.....	2
Fig 1.3	Doppler ultrasound block diagram and wave forms.....	2
Fig 1.4	Time varying velocity spectrum.....	3
Fig 1.5	Zoomed systole to show pixel brightness.....	3
Fig 2.1	Doppler sonogram is the lower portion of the image.....	5
Fig 2.2	Doppler image with lines words and numbers removed.....	6
Fig 2.3	A single profile from a downward vertical scan through an image.....	7
Fig 2.4	Same profile with noise removed.....	7
Fig 2.5	Next single profile from a vertical scan.....	8
Fig 2.6	Same profile with noise removed.....	8
Fig 2.7	Two clicks made on at the beginning and end of a cycle in an image.....	8
Fig 2.8	Envelopes resulting from threshold evaluation on a series of profiles.....	9
Fig 2.9	Envelopes for a different images series of profiles.....	9
Fig 2.10	Smoothed envelopes for the envelopes in fig 2.8.....	9
Fig 2.11	Smoothed envelopes for the envelopes in fig 2.9.....	9
Fig 2.12	Edge detection of a Doppler image using the Sobel method.....	10
Fig 2.13	Doppler image converted to frequency space.....	10
Fig 2.14	Gaussian filter applied to frequency space image.....	11
Fig 2.15	Edge image with a Gaussian filter threshold of 50.....	11
Fig 2.16	Envelope curves as a result of threshold evaluation of an edge image.....	12
Fig 2.17	Smoothed envelopes for the envelopes in fig 2.16.....	12
Fig 2.18	Edge image using a Gaussian threshold of 35.....	12
Fig 2.19	Smoothed envelopes from edge image with threshold of 35.....	13
Fig 2.20	Least squares curve fitted to a differential method profile.....	13
Fig 2.21	First differential plot of the Least squares curve profile in fig 2.20.....	13

Fig 2.22	Least squares curve fitted to the next profile using the differential method....	14
Fig 2.23	First differential plot for the profile in fig 2.22.....	14
Fig 2.24	New image to use the differential threshold on.....	14
Fig 2.25	Unsmoothed envelopes for a cycle in the image of fig 2.24 using the differential method.....	15
Fig 2.26	Smoothed envelopes for fig 2.25.....	15
Fig 2.27	A profile to use the differential method on between two clicks.....	15
Fig 2.28	Summed pixel value versus vertical scan position for the profile in fig 2.27..	15
Fig 2.29	Summed pixel value as a percentage of the total value for a vertical scan.....	15
Fig 2.30	Image with five cycles to try the percentile method on.....	16
Fig 2.31	Smoothed maximum velocity envelope for a cycle in fig 2.30.....	16
Fig 3.1	Least squares fit to an envelope using three components of a Fourier series.	19
Fig 3.2	Least squares fit to an envelope using fifteen components of a Fourier series	19
Fig 3.3	Image showing reverse flow and two clicks made on the image.....	20
Fig 3.4	Maximum and minimum velocity envelopes of a cycle in fig 3.3.....	20
Fig 3.5	Envelopes from fig 3.4 merged to form one envelope with fitted curve.....	20
Fig 3.6	Image with very little reverse flow.....	21
Fig 3.7	First Crossover point on the envelope curve (23).....	22
Fig 3.8	Maximum velocity envelope is used to fit a curve if reverse flow is not needed	23
Fig 2.9	Merged envelopes used to fit a curve if reverse flow is needed.....	23
Fig 2.10	Average pixel value versus threshold for a cycle in a noisy image.....	24
Fig 3.11	Area ratio versus threshold for a cycle in a noisy image.....	24
Fig 3.12	Average pixel value versus threshold for a cycle in an image with little noise	25
Fig 3.13	Area ratio versus threshold for a cycle in an image with little noise.....	25
Fig 3.14	An image with cycles that vary in pixel brightness.....	25
Fig 3.15	Average pixel value versus threshold for five cycles in a noisy image.....	26
Fig 3.16	Area ratio versus threshold for five cycles in a noisy image.....	26

Fig 4.1	Image showing four cycle that can be used to find an average area ratio for the image.....	29
Fig 4.2	Zoomed cycle in fig 4.1 showing approximate positions of three clicks....	29
Fig 4.3	Area ratios for four cycle of fig 4.1 over twenty trials (three clicks).....	30
Fig 4.4	Mean area ratios for the twenty cycles (top – four cycles, bottom – three cycles)	30
Fig 4.5	Envelope curve and area ratio for the first cycle in fig 4.1.....	31
Fig 4.6	Zoomed view of fig 4.5 showing region between systole and diastole.....	31
Fig 4.7	Area ratios for four cycle of fig 4.1 over twenty trials (two clicks).....	32
Fig 4.8	Mean area ratios for the twenty cycles for two clicks (top – four cycles, bottom – three cycles).....	32
Fig 4.9	Envelope curve showing a local valley in systole.....	33
Fig 4.10	Zoomed portion of fig 4.9 showing the local valley in systole.....	33
Fig 4.11	Envelope curve showing a bent systole but no local valley.....	33
Fig 4.12	Zoomed portion of the systole for fig 4.11. A local valley is present in the fitted curve of the envelope.....	33
Fig 4.13	Envelope curves for thresholds 10 to 40 for cycle one in fig 4.1.....	34
Fig 4.14	Envelope curves for thresholds 50 to 100 for cycle one in fig 4.1.....	35
Fig 4.15	Average pixel versus threshold and area ratio versus threshold for cycle one in fig 4.1.....	36
Fig 4.16	Changes in systolic and diastolic peak values for thresholds 40 to 100....	36
Fig 4.17	Summary of ave. pixel value and area ratio vs threshold for four cycles in fig 4.1 using the ten-pixel-average program.....	37
Fig 4.18	Edge detection of cycle one in fig 4.1. Thresholds 30 to 80.....	38
Fig 4.19	Smoothed envelopes for edge detection method of cycle one in fig 4.1 thresholds 30 to 80.....	39
Fig 4.20	Area ratio and average pixel value versus threshold for cycle one using the edge method.....	40
Fig 4.20a	Maximum velocity envelopes for the ten-pixel-average (threshold=70) and edge (m=50) graphed together.....	40
Fig 4.21	Area ratio and average pixel value versus threshold for all four cycles in fig 4.1 using the edge method.....	41
Fig 4.22	Smoothed envelope curves for cycle one in fig 4.1 using the differential method (thresholds 500 to 3500).....	42

Fig 4.23	Area ratio and average pixel value versus threshold for cycle one in fig 4.1 using the differential method.....	43
Fig 4.24	Area ratio and average pixel value versus threshold for all four cycles in fig 4.1 using the differential method.....	44
Fig 4.25	Envelope curves for cycle one in fig 4.1 using the percentile method (thresholds 5% - 30%).....	45
Fig 4.26	Envelope curves for cycle one in fig 4.1 using the percentile method (thresholds 35% - 50%).....	46
Fig 4.27	Average pixel value and area ratio versus threshold for percentile for cycle one in fig 4.1.....	46
Fig 4.28	Average pixel value and area ratio versus threshold for all four cycles in fig 4.1	47
Fig 4.29	Noisy image with dim pixel values.....	48
Fig 4.30	Maximum velocity envelopes using the ten-pixel-average method for thresholds 40-90 on a noisy dim image.....	49
Fig 4.31	Average pixel value and area ratio versus threshold for ten-pixel-average Method on a noisy dim image.....	49
Fig 4.32	Edge images for thresholds $m=30$ and $m=40$ using the edge detection method on the image in fig 4.29.....	50
Fig 4.33	Maximum velocity envelopes for thresholds $m=30$ and 40 for the images in fig 4.32.....	50
Fig 4.34	Edge image for $m=20$ and maximum envelope for this threshold.....	51
Fig 4.35	Maximum velocity envelopes for thresholds 1000 to 2500 for the differential method on fig 4.29.....	52
Fig 4.36	Maximum velocity envelopes for thresholds 5% and 10% for the percentile method on fig 4.29.....	52
Fig 4.37	Area ratio versus image number for ten-pixel-average and edge methods on a complete image set (person 6).....	54
Fig 4.38	Area ratio versus image number for differential and percentile methods on a complete image set (person 6).....	54
Fig 4.39	Loupas ratio versus image number using ten-pixel-average method on A complete image set (person 6).....	54
Fig 5.1	Graphs showing curve fits for 5, 20 and 50 Fourier components and area ratio versus number of components.....	56
Fig 5.2	Area ratio plotted against image number for person 15.....	58
Fig 5.3	Area ratio, heart rate, blood pressure and nerve activity plotted against image Number for person 15.....	58

Fig 5.4	Doppler images and waveforms for person 15 before tilt and at syncope....	59
Fig 5.5	Area ratio (A), blood pressure (B), heart rate (H) and nerve activity (N) for person 22.....	60
Fig 5.6	Doppler image and maximum velocity envelope at point 13 (faint) for Person 22.....	61
Fig 5.7	Area ratio versus image number for person 1.....	61

GLOSSARY

Baroreceptors	Nerves responding to pressure changes
Matlab	Scientific analysis software
Maximum velocity envelope	Curve joining maximum velocities in systole and diastole
MSNA	Muscle sympathetic nerve activity
Diastole	Dilation of the heart chambers so they fill with blood
FFT	Fast Fourier transform
Least squares	Optimization method of curve fitting
Loupas Index	Ratio using Fourier amplitudes for a complete systole/diastole cycle and average envelope velocities in diastole
Pixval	Command in matlab that enables pixel intensity to be displayed
RGB	Red, green, blue (pixel brightness)
SMA	Superior mesenteric artery – feeds blood to the intestines
Sobel	Method of finding image edges using derivatives
Sonogram	Doppler ultrasound image showing a velocity spectrum for cycles of systole/diastole
Syncope	Fainting
Systole	Contraction of the heart forcing blood into the aorta
Ten-pixel-average	Value for each point of a profile looking ten pixels ahead and averaging

CHAPTER ONE

Background

1.1 Vasovagal syncope

Vasovagal syncope is not a serious or life threatening condition, but in effect an abnormal reflex. This results in a drop in blood pressure leading to decreased blood flow to the brain resulting in dizziness or fainting. The mechanism of vasovagal syncope is the subject of a great deal of research.

When we sit or stand, blood settles in the legs and abdomen. As a result, less blood returns to the heart. The blood vessels leaving the heart have detectors in them called baroreceptors that detect a decrease in blood pressure. The baroreceptors send a message to the brain, which in turn sends a signal to the heart to increase the heart rate, and tighten up the blood vessels. This process occurs constantly in all of us as we adapt to changes in posture. In vasovagal syncope, an abnormal reflex occurs that results in withdrawal of the message that speeds up the heart and tightens up the vessels, often because of an overshoot in the reflex that compensates for the fall in blood pressure. The resultant decrease in blood flow to the brain will result in dizziness or light-headedness if mild, and progress to fainting or loss of consciousness if more severe.



fig 1.1 Tilt table at Christchurch hospital

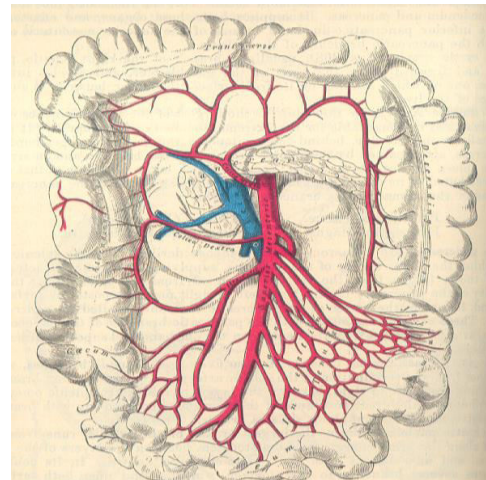


fig 1.2 Mesenteric artery page 554 Gray's Anatomy, Classic Collectors Edition

Tilt testing is performed to try to reproduce an episode. Tilt testing involves a postural stress test. Blood pressure, heart rate, nerve activity and measures of the superior mesenteric artery (feeds blood to the intestines) Doppler waveform are monitored with the subject lying horizontal, and tilted to 60° to try to trigger an episode of loss of consciousness.

1.2 Doppler ultrasound

Doppler ultrasound is a technique for making non-invasive velocity measurements of blood flow. Ultrasound is transmitted into a vessel and the reflected ultrasonic echoes from blood particles are detected. Because the blood is moving, the sound undergoes a frequency shift that is described by the equation:

$$\Delta f = \frac{2fv\cos(\theta)}{c} \quad (1)$$

v - velocity of the blood

f - transmitted frequency

θ - Doppler angle

c - speed of sound

Δf - Doppler shift frequency

v is directly proportional to Δf

The transmitted frequency is usually in the range 2 – 10 MHz and the Doppler shift frequencies, which are a small fraction of the transducer frequency, is in the audible range. In a pulsed Doppler device there is only one crystal. This crystal serves both as a transmitter and receiver. It transmits a short ultrasonic impulse lasting for a fraction of a millisecond. Then it switches to receive mode. A 'time gate' can be adjusted so that only echoes from a certain depth can be detected. Thus pulse Doppler enables the device to target the mesenteric artery over a number of other arteries in the vicinity.

The simplest method to determine the Doppler shift is to make use of 'beats'. When two signals of similar frequency are added together, beats are formed due to the signals gradually varying between being in and out of phase and hence alternately reinforcing and cancelling. Thus, if the signal due to echoes from moving structures is added to a reference signal at the original frequency, the beat frequency is the Doppler shift frequency Δf . Usually there is a range of velocities in blood vessels and a spectrum of frequencies make up the Doppler signal – see fig 1.3.

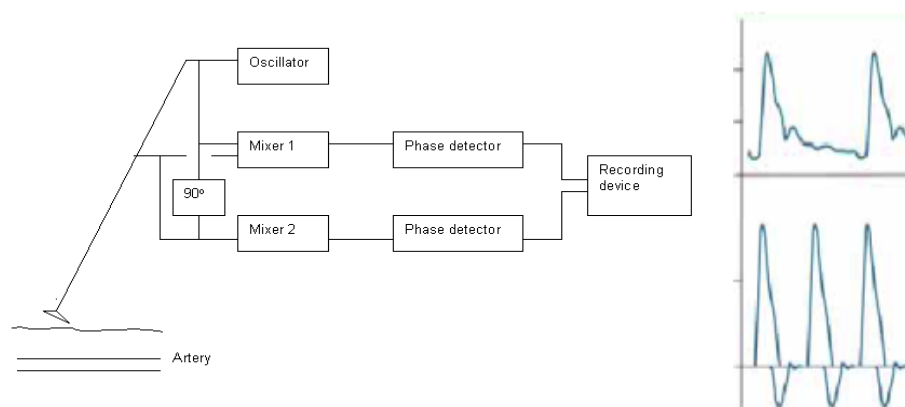


fig 1.3 Doppler ultrasound scanning with velocity profile. Top right – velocity waveform for cerebral artery. Bottom right velocity waveform for peripheral artery.

Even at one time point and at a given position in a blood vessel, there is a range of velocities. The maximum velocity is in the centre of the blood vessel with the minimum velocity near the vessel wall. Therefore, the waveforms diagrammed in

fig 1.3 are not measured as clean lines. Instead, the lines are spread over a range of velocities, ranging from the maximum velocity to velocities that are typically close to zero.

The goal of this project is not to simply find the average velocity, but to attempt to map out the maximum speed (absolute value of the velocity) of the blood flowing in the vessel. Trying to map the maximum speed of the blood can create continuous and non-continuous plots as show in the right-hand side of fig. 1.3. This requirement to find the maximum speed of the blood flow is why we decided that uniquely designed analysis methods might work better or faster than existing image enhancement techniques.

In addition to the range of blood speeds measured, the ultrasonic Doppler signal is often filled with noise, too. Spurious signals can create a “snow” of noise on the image that needs to be ignored. We explain these features below.

1.3 Indexes

Fig 1.4 shows a typical display of a time varying velocity spectrum. The taller systolic region corresponds to heart contraction forcing blood into the aorta; followed by the diastolic region, which corresponds to dilation of the heart chambers allowing them to fill with blood. The cycles are repeated as the heart beats rhythmically. The horizontal axis measures time, the vertical axis measures velocity and the brightness of pixels in the systolic/diastolic regions is proportional to the number of blood cells with a particular velocity at a particular time (see fig 1.5).

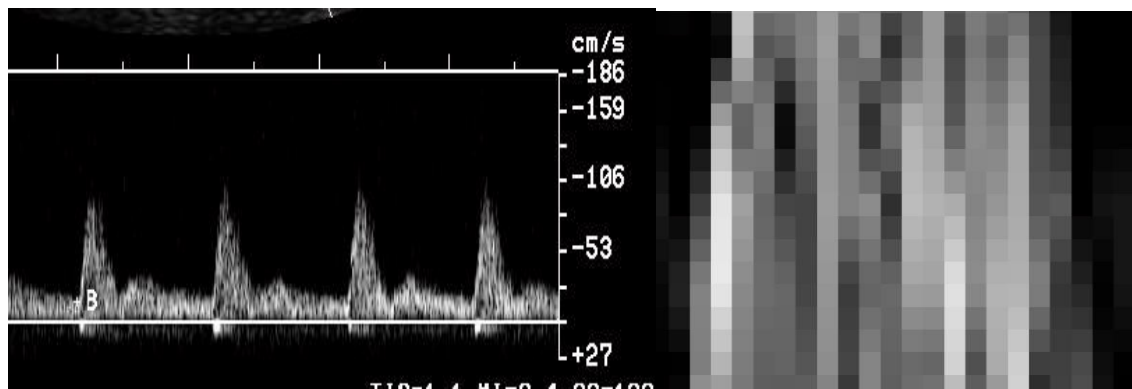


fig 1.4 Display of time varying velocity spectrum

fig 1.5 Zoomed systole to show pixel brightness

In general cerebral blood vessels are not subject to muscular resistance and for this reason these vessels are characterized by relatively high diastolic flow (fig 1.3 top right). On the other hand blood flow cannot flow into peripheral arteries in the rest of the body (mesenteric artery for example) unhindered so the diastolic flow is relatively low (fig 1.3 bottom right)¹. Changes on posture can alter the diastolic flow in arteries².

Both the audio signal and the digital image can be used to analyse Doppler shift information in systole and diastole. The audio output allows an operator to easily differentiate laminar from turbulent flow. Laminar flow produces a smooth, pleasant tone because of the uniform velocities. Turbulent flow, because of the presence of many different velocities, results in a commonly high-pitched and whistling or harsh

and raspy sound. The trained ear can readily appreciate minor changes in spectral composition more readily than the eye. The major limitations of audio Doppler outputs is the requirement for subjective interpretation and the lack of permanent objective records. To objectively study blood flow (particularly changes in diastolic flow) analysing digital image records is the obvious choice.

An Index often used, as an indicator of changes in arterial blood flow, is the S/D ratio. It takes the peak systolic velocity V_s and divides it by the end Diastolic velocity V_d .

$$S/D = \frac{V_s}{V_d} \quad (2)$$

An image may have 4 to 8 Systolic/Diastolic cycles so an average found for the S/D ratio for the image. $(S/D)_{ave}$ can then be compared to other images in a sequence of scans.

However, in the range of images scanned for this study, many are like the one in fig 2.1. It has eight cycles with end Diastolic velocities near zero for some time before Systole starts again. The above ratio would become a very large number!

To achieve more discrimination from cycle to cycle and image to image a better measure such as the ratio of the area under the curve in systole versus that in diastole, using a maximum velocity envelope, should describe more completely the changes taking place in diastole and hence give a measure of vasoconstriction/vasodilatation of the superior mesenteric artery.

CHAPTER TWO

Threshold methods

2.1 Preparing Images

Matlab (MathWorks, Massachusetts, USA) downloads images from a CD containing anonymous ultrasound TIFF files (General Electric, Diagnostic Ultrasound, Logiq 7, Giles, UK). There are forty people in the study and each person may have up to 80 images. Individuals participating in this project have given informed consent and the project approved by the local ethics committee. People in the study are anonymous and have been called person1, person 2 etc and images for a particular person labelled from 1 to 80.

It is the lower Doppler portion of the image that is needed for the study. Some aspects of the top part of the image are in colour so the image has to be changed to gray-scale; then Matlab has one number for pixel values instead of 3 RGB values. Each image has five or more cycles of systole and diastole (high peak, small peak). These cycles may vary considerably from cycle to cycle and image to image for a particular person.

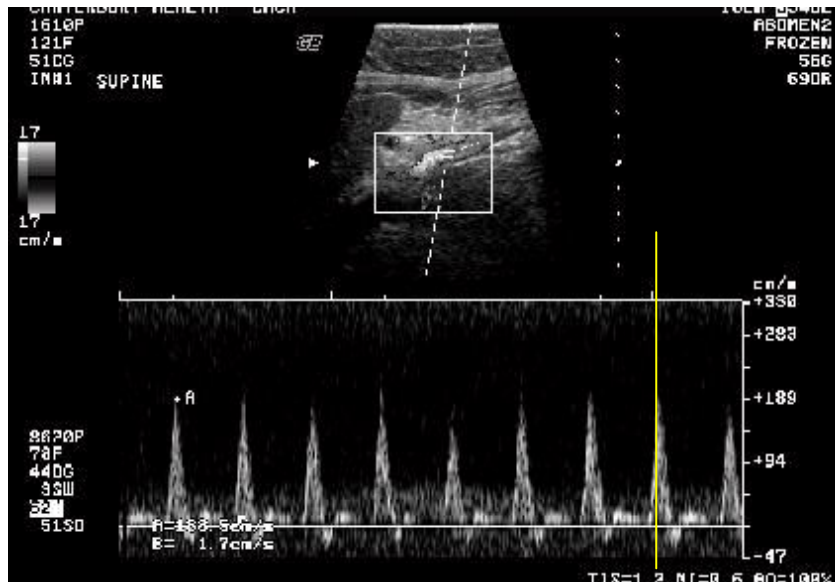


fig 2.1 The lower portion of the ultrasound image (with the two reference lines) is needed for the study.

The lower portion of fig 2.1 shows the familiar systolic and diastolic pulse sequences. This image shows considerable ‘noise’ below the top reference line and above the small diastolic peaks. Most cycles in this image have bright reverse flow pulses below the bottom reference line (zero velocity). In general levels of noise in parts of images, large variations in brightness within an image and cycles missing make it nearly impossible to automate the process of finding a maximum velocity envelope (Appendix One shows the image set of person 6 in the study).

Instead of Matlab scanning the whole Doppler portion and finding maximums and minimums, with suitable code ³, it is faster to interactively choose one complete systolic/diastolic cycle from the image thus avoiding poor quality cycles. Then the procedure can be repeated for other good cycles in that image. Repeated measurements of the data are then available if averaging is required.

Matlab's 'impixel' command enables the user to point the cursor and "click" the mouse (generically termed "click" throughout this work) with the cursor on any part of the image and thus the coordinate positions to be stored and used in the main program. A click can be made at the beginning of a cycle (just before systole) and at the end of the cycle (just after diastole). A third click can be made between systole and diastole or code to locate that point automatically written. Errors due to clicking will be explored in chapter four.

Next, the two reference lines in the Doppler sonogram have to be detected. The top reference line rarely exceeds 330 cm s^{-1} and the bottom line represents zero velocity. Matlab can begin a series of adjacent vertical scans, positioned in the image to ensure they will always pass through the Doppler reference lines. Their pixel value is 255 and their vertical position can change from image to image. Five vertical scans have proved enough starting from a safe position above the top line (bearing in mind that this line can slightly move and checking with a good cross-section of the images). It is unnecessary to cover the full vertical length of the image, as this will add to the total scan time. If Matlab detects that for five vertical scans, corresponding y positions all have pixel values of 255 then it will store those y positions. These y positions correspond to the reference lines. Note that the printing and numbers in the image also have a pixel value of 255. Matlab code can now be used to eliminate these lines and printing as in fig 2.2.

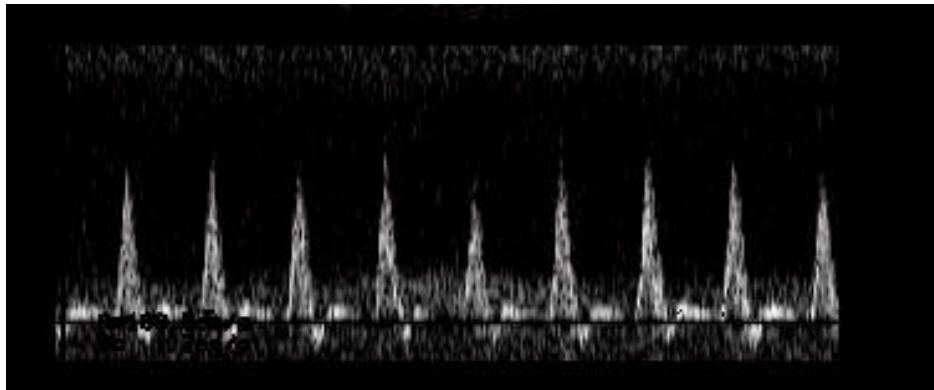


fig 2.2 Doppler image with lines, writing and numbers removed.

Comparing fig 2.2 with fig 2.1, the systolic and diastolic pulses have not been degraded by removing pixel values of 255 except a little on the left of the image. Removal of the zero velocity reference line has left a horizontal gap. It can now be 'filled' by making it the average of the pixel value above and below for each x position. This procedure is particularly valuable if using an edge filter on the image.

The image is now ready to do a series of vertical scans for a particular cycle in the image. A first click is manually made before systole and a second click made after diastole. Every time-position (x position) between these clicks is vertically scanned and a profile recorded. Fig 2.3 is an example of a single vertical profile. The vertical line in fig 2.1 through a systolic region on the right shows the actual region of the vertical scan (top to bottom). It is obvious in this profile where the small noise peaks are and the point where the systole begins (near 130). This point corresponds to the maximum velocity. The small noise peaks in the beginning of the profile correspond to the noise just below the top reference line in the original image.

There are various methods that can be used to determine the maximum velocity point in each profile: Pixel changes from an edge filter, a percentile, method, a differential

method and a method that for each position in the profile, the program looks ahead ten pixel spaces and calculates the average pixel value over those ten spaces. This last method I will call the ten-pixel-average method.

2.2 Ten-pixel-average Method

The ten-pixel-average method determines the maximum velocity point in a profile by systematically moving through a profile and for each point looking ahead ten spaces and calculating the average pixel value over those ten spaces (a 10 pixel running average). The graph in fig 2.3 shows a profile, and Matlab checks the profile from left to right. For each position below about 130 the average pixel value ten spaces ahead for each position will be low compared to after 130 when it rises steeply.

As an example: At position 100 on the profile, the next ten spaces from here have an average pixel below 5. However at position 150, the next ten spaces have an average pixel value over 100.

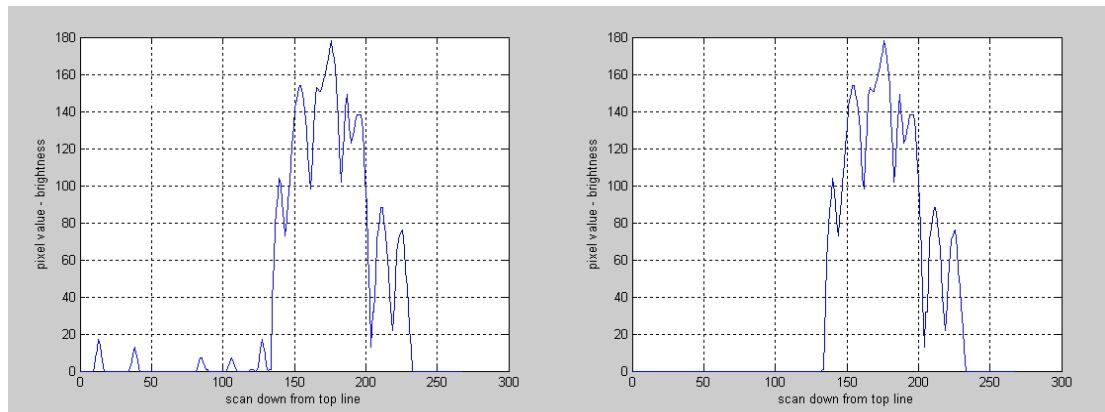


fig 2.3 Single improfile scan downwards. **fig 2.4** Same scan, noise removed.

A threshold average pixel value over the next ten spaces can be set so that the movement through the profile stops if the threshold is exceeded. Compensation can be made for the fact that the stop will occur just before the point of sharp increase due to a rise in average pixel value before the actual point. For the profile in fig 2.3, if the average pixel value threshold is chosen to be 20 then the check through the profile will stop just before 130 and this is indicated in fig 2.4 by the small noise peaks having been removed.

There will of course be many profiles as a result of consecutive vertical scans between the two clicks. The graph in fig 2.5 shows the next profile in the series. The initial noise peaks are a little higher so a larger threshold (30) will be necessary to make the check through the profile stop again around 130. This is indicated in fig 2.6 with the noise removed.

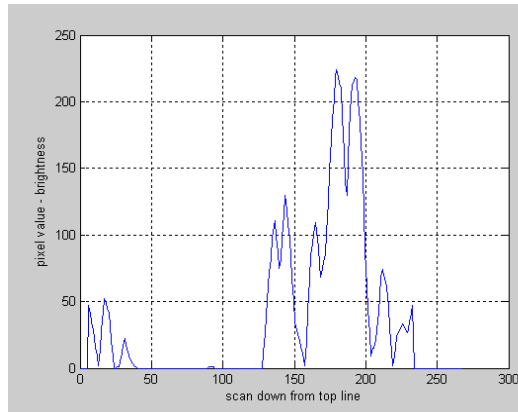


fig 2.5 Next single improfile scan

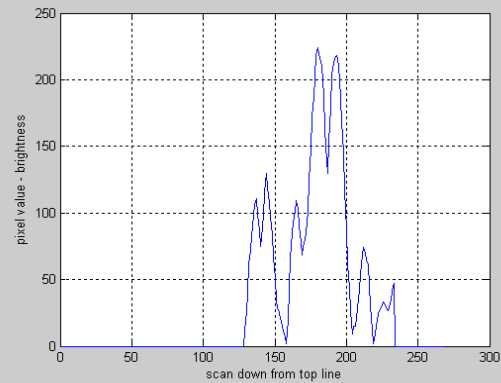


fig 2.6 Same scan, noise removed

Choosing a threshold then becomes a matter of viewing all the images belonging to a particular person and a judgement made to ensure the worst noise can be removed. However it is necessary to mention that choosing a threshold too high will result in the stop point being located beyond point 130 in this example. In other words the program will miss the maximum speed and select a slower speed. The effect of various thresholds will be further investigated in chapter 4.

If now the mouse is used to click at the beginning and end of a cycle as in fig 2.7 (image on full screen to make the clicks), and the ten-pixel-average program used for all profiles between the clicks, a maximum velocity envelope results which can be plotted. This plot is shown in fig 2.8 (top curve). In fact each profile can be checked in reverse to get the minimum velocity envelope also (lower curve).

Fig 2.9 shows the maximum and minimum velocity curves after the procedure is used on a cycle in another image. It is evident that in general the envelopes of scanned cycles in images may contain spikes due to sharp changes in the region of interest, or noise in the image. Before finding areas within these curves it will be necessary to smooth the envelope without reducing its ability to represent the cycle.

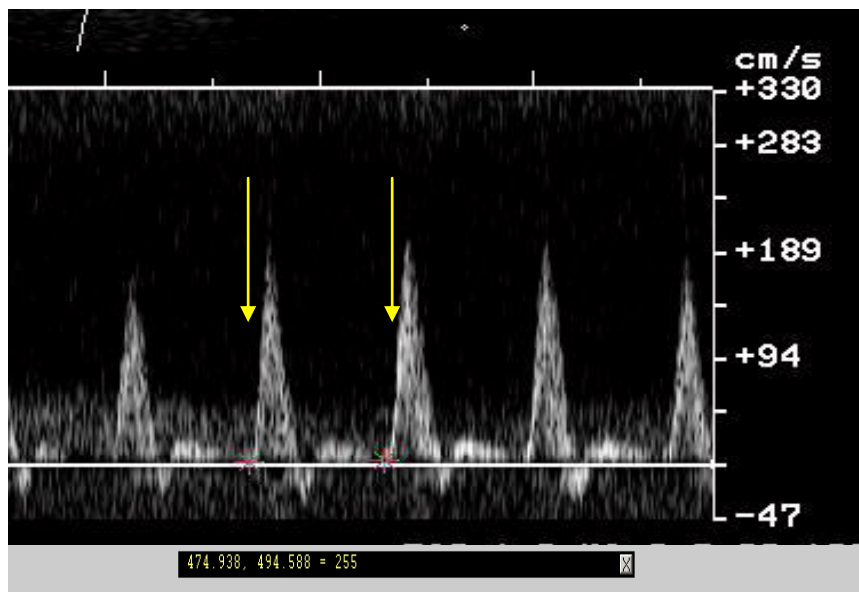


fig 2.7 Two clicks on a cycle – beginning and end. (pixval display at the bottom)

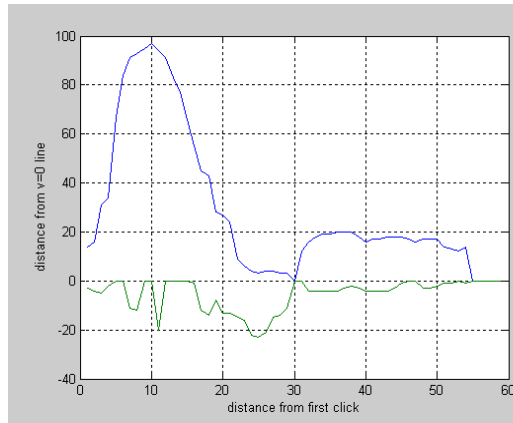


fig 2.8 Result of a series of vertical scans with threshold points marked

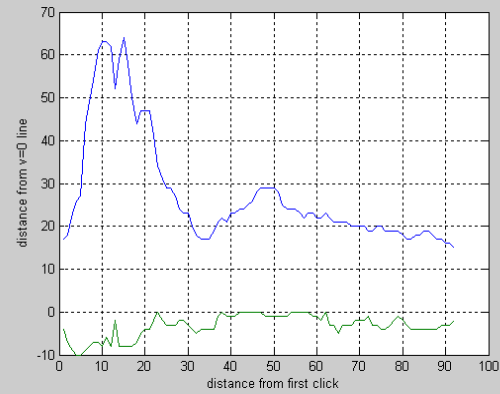


fig 2.9 Different image same procedure

The procedure for smoothing the curve involves looking at each point on the envelope curve and comparing it with the points either side. The following is the code used in Matlab.

```
for k=1:50
for j=1:length(F)
if ((j>=2)&(j<=length(F)-1)&(abs(F(j)-(F(j-1)+F(j+1))/2)>51-k))
F(j)=(F(j-1)+F(j+1))/2;
end;end;end
```

The first and last points in the envelope are left out in this process. The process checks the envelope points 50 times. In the first run, if a point is greater than 50 times larger in absolute value than the average of two points either side (a very sharp peak), then it becomes the mean of those two points. This process is then systematically repeated, dropping the difference between the peak and the average of the two points either side by one until no more change takes place. The smooth curve in fig 2.10 shows the effect of this code on the same region as fig 2.8. The smoothed envelope curve now looks like the cycle clicked on in fig 2.7 for the threshold chosen. The height of the systole is just over 90 and this compares well when using the Matlab command 'pixval' on the original image cycle and dragging the mouse up from the zero velocity line to the systole peak (110). Fig 2.11 shows the effect of smoothing the envelopes in fig 2.9.

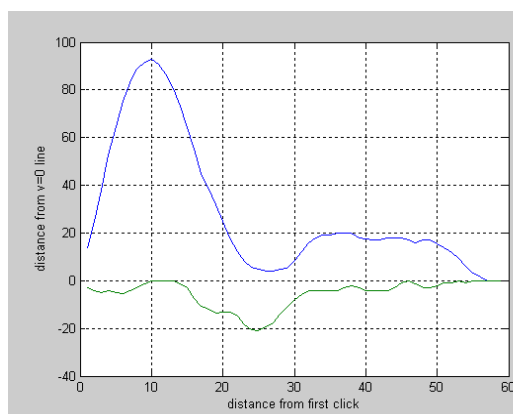


fig2.10 Smoothed curve for fig 2.8

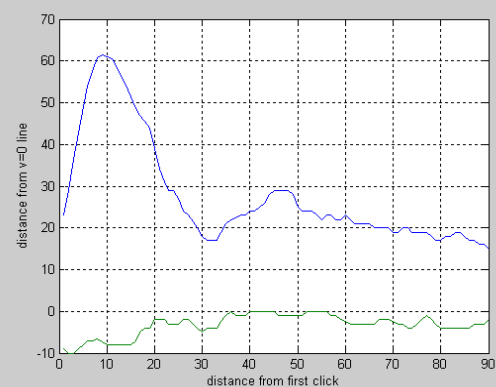


fig 2.11 Smoothed curve for fig 2.9

2.3 Edge Filter Method

Another method of finding the maximum velocity envelope is to use an edge filter on the image such as Sobel. The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of image points is a maximum. Fig 2.12 shows an edge detection image using the Sobel command in Matlab on the image in fig 2.1.

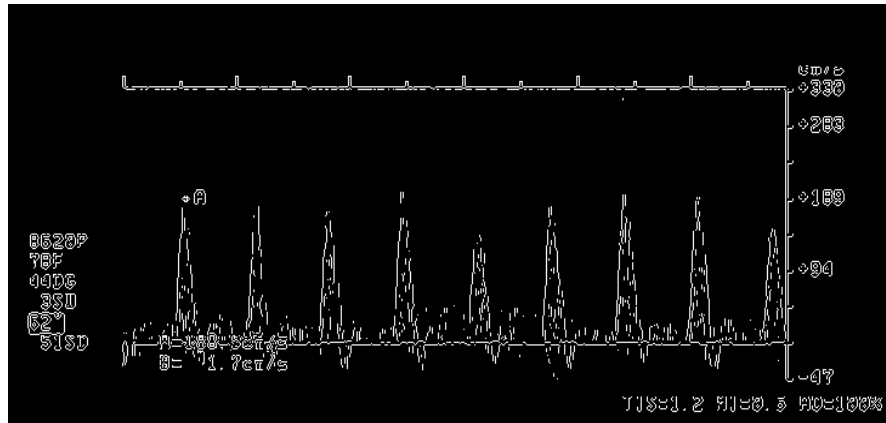


fig 2.12 Edge detecton of a Doppler image using the Sobel method

The code to initiate Sobel in Matlab is: `d=edge(c,'sobel',(graythresh(c) * 0.1),'both')`. There is a threshold value in this code that can be changed to alter the edge detection. In this case the threshold is 0.1. However changing the threshold from 0.1 to say 0.2 drastically changes the edge detection for these ultrasound images to something unusable. It is then preferable to use another method to finely adjust the edge detection process.

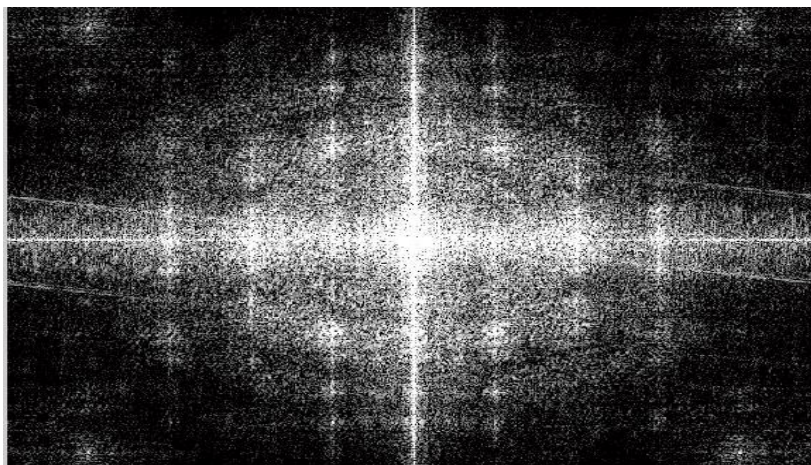


fig 2.13 The image in fig 2.13 converted to frequency space – high frequency further out.

This next step may seem contradictory to edge detection but it makes possible the fine adjustment required. First the original image is transformed into frequency space using a Fast Fourier Transform. The low frequency data is in the middle and high frequency further out to the sides as in fig 2.13. (FFT of fig 2.1)

By removing data from the outside – high frequency – the original image will lose some of its boundary definition as seen when the inverse FFT is applied. This can be done for example using a Gaussian mask as in fig 2.14. The outer data is progressively eliminated when it is applied to the FFT of the image. The equation of the Gaussian mask takes the form: ⁴

$$\frac{\exp(x^2 + y^2)}{2m^2} \quad (3)$$

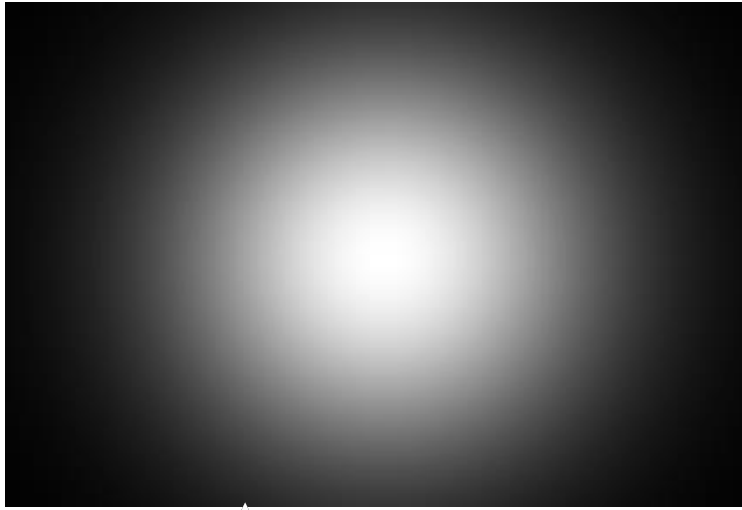


fig 2.14 Gaussian mask to remove high frequency – size can be changed.

The value of m determines the fall-off or ‘size’ of the Gaussian mask. If for example the value of m is 50 and this is applied to the FFT, then when the inverse FFT is performed to get back the image the edges will be slightly blurred (convolution is an alternative). Now the Sobel edge detector is applied to this version of the original image and the result is fig 2.15. Both systole and diastole are now well defined with some noise.

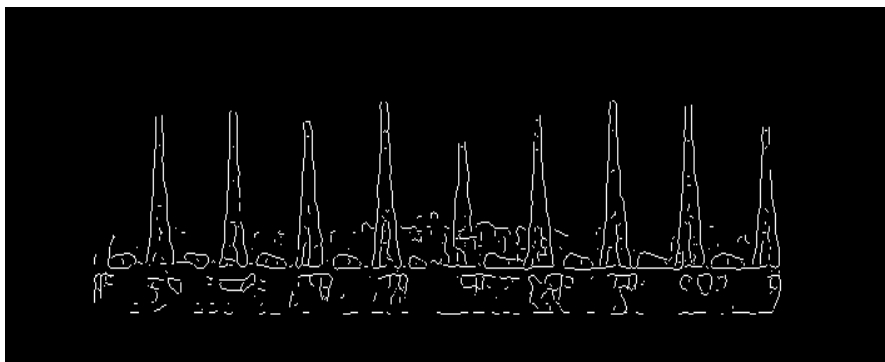


fig 2.15 Gaussian filter + edge filter m=50

The pixel values are now 1 or 0 after edge detection. So the next part of the code is easily written to detect the change from 0 to 1 provided the threshold for m has removed noise in two or three of the cycles. Again a series of vertical scans between

two clicks (enclosing systole and diastole) detecting the 0 to 1 pixel change will yield the maximum and minimum velocity envelope after checking profiles in both directions. Fig 2.16 shows the envelope for the second complete pulse from the right in fig 2.16 and if this is smoothed (using the same code as the ten-pixel-average program) the curve is shown in fig 2.17.

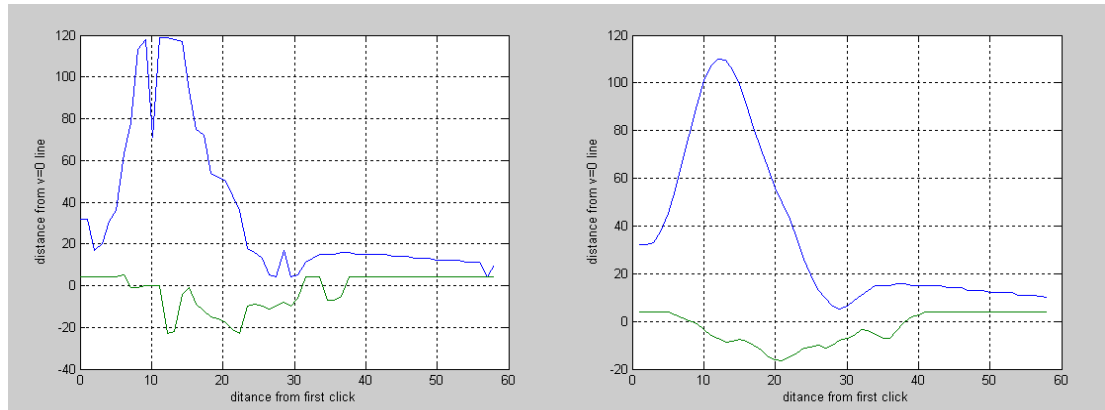


fig 2.16 unsmoothed envelope for second second cycle to the right in fig 2.15 **fig 2.17** smoothed envelope for second cycle to the right in fig 2.15.

If a threshold of $m=35$ is used the edge detection shows more cycles without noise as in fig 2.18. However the tops of the systole tend to be open which can cause spikes when detecting changes from 0 to 1 in pixel value.

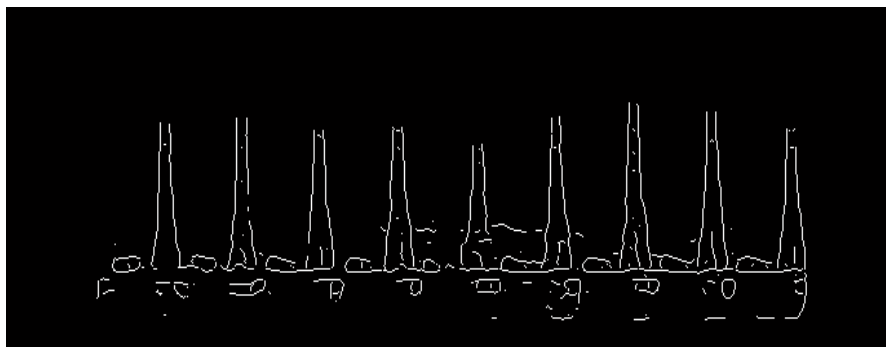


fig 2.18 Edge filter result with threshold on 35

The smoothed curve is however much reduced as in fig 2.19 when compared to the envelope in fig 2.17.

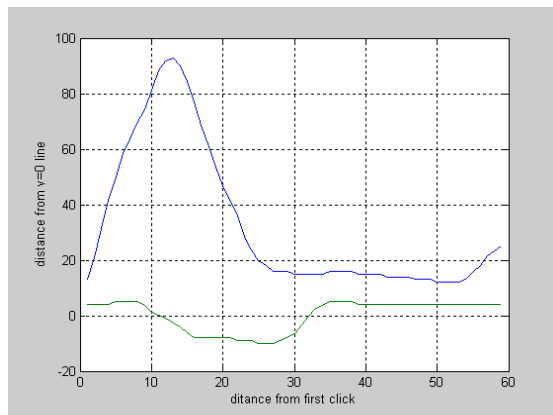


fig 2.19 Smoothed result of scanning the edge image in fig 2.18 with threshold of 35.

2.4 Differential Method

Another method of finding the maximum and minimum velocity envelope is to fit a curve using Least Squares to each profile scanned between the two clicks. Although it is hard to see the profile in fig 2.20 has been fitted with a tight fitting curve. The Least Squares method in this case used 50 components in a Fourier Sine series (some high frequency data are lost):

$$a_1\sin(x) + a_2\sin(2x) + a_3\sin(3x) + \dots \quad (4)$$

The method finds the coefficients a_1 , a_2 , a_3 etc and thus Matlab now has the parameters for the tight fitted curve and takes the first differential of the curve. Fig 2.21 shows the plot of the first differential for the profile in fig 2.20.

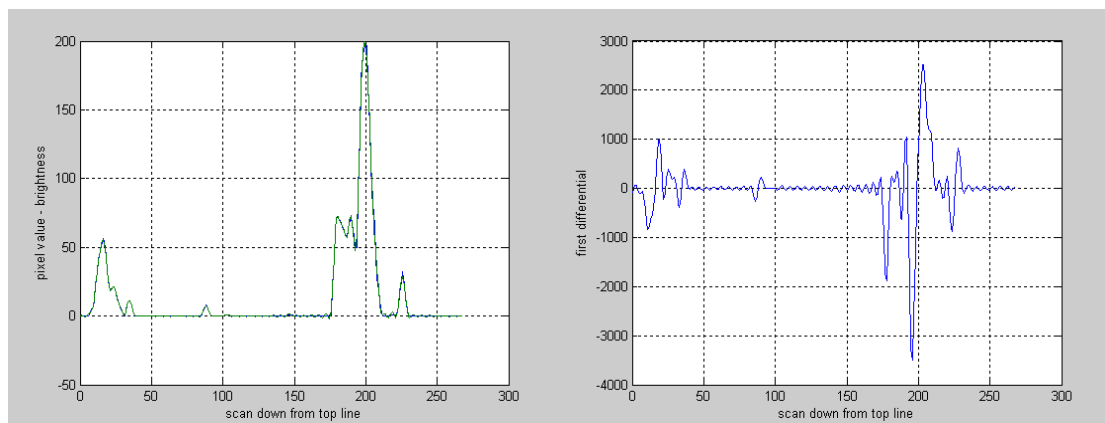


fig 2.20 Single improfile scan with tight fitting curve

fig 2.21 First differential plot of tight fitting curve in fig 2.20

The initial peaks in fig 2.21 are noise and are less than or equal to 1000. So if a threshold is to be taken using the first differential it should be greater than 1000 (absolute value). The next two diagrams – fig 2.22 and fig 2.23 show another profile with tight fitting curve and differential plot.

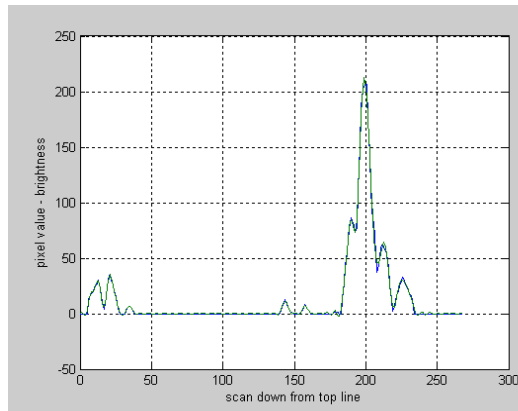


fig 2.22 Next single improfile scan with tight fitting curve

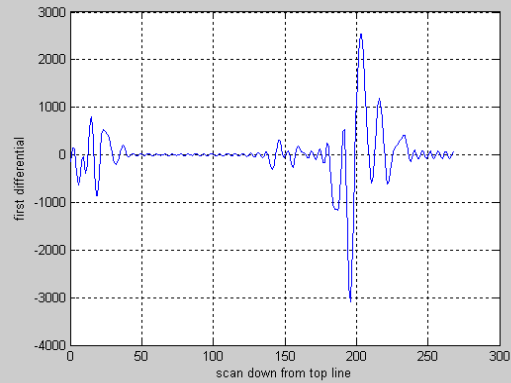


fig 2.23 First differential plot of tight fitting curve in fig 2.22

If in the image in fig 2.24, clicks at the beginning and end of the second complete cycle on the right are made, then each profile checked for the first differential first threshold a maximum and minimum velocity curve results as in fig 2.25 (Threshold = 1000). Some large spikes are present in these envelope curves so by using the smoothing code as in the ten-pixel-average method the plot in fig 2.26 results. Use of pixval on the image shows the systolic height to be ~80 and the envelope curve to be near 70. Initial noise differential peaks vary considerably from profile to profile. Threshold changes are then usually by hundreds (cf. Sobel).

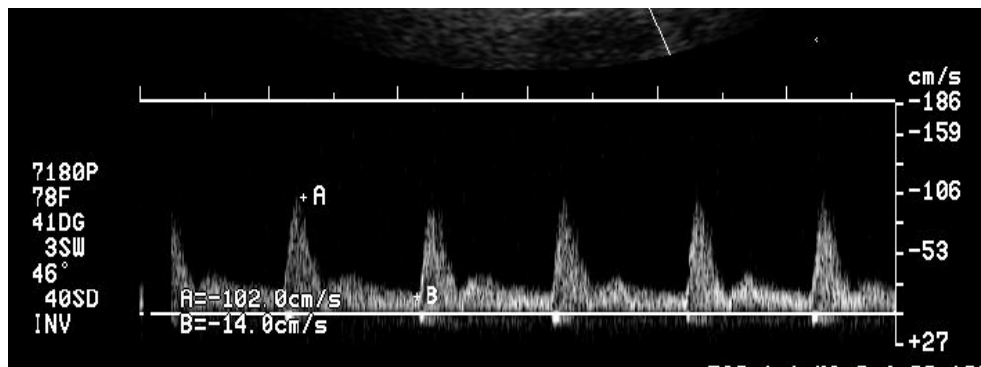


fig 2.24 New image to use the differential threshold on.

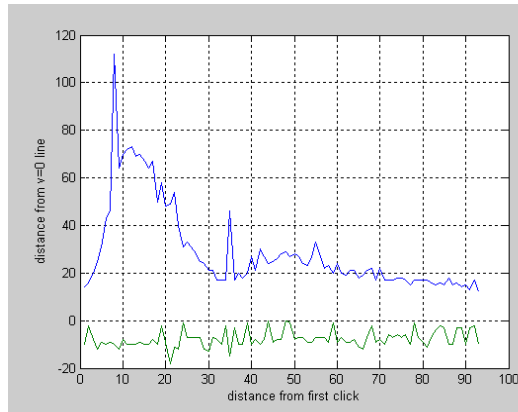


fig 2.25 Unsmoothed envelope th=1000

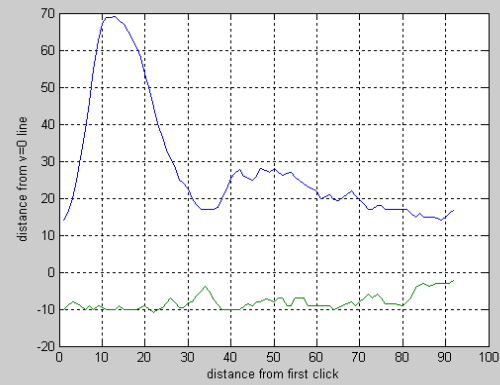


fig 2.26 Smoothed envelope th=1000

2.5 Percentile Method

The percentile method ¹ involves summing the pixel values in each profile between the two clicks as the profiles are checked in both directions. In fig 2.27 the pixel values are summed as we move right to left and shown in fig 2.28. This is converted to a percentage of the total value in fig 2.29.

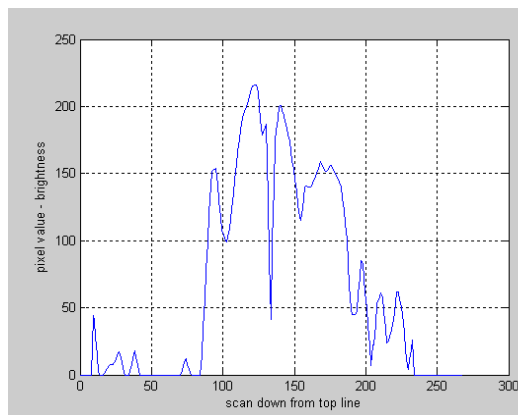


fig 2.27 One profile from a series between two clicks

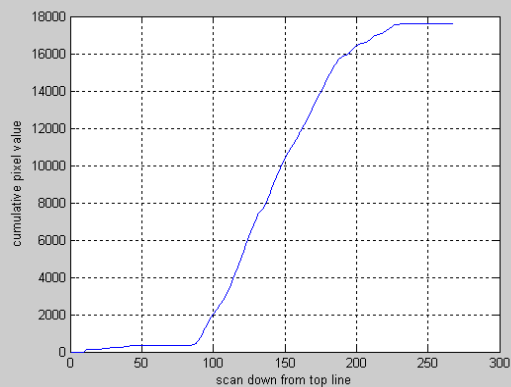


fig 2.28 Summed pixel values from fig 2.27

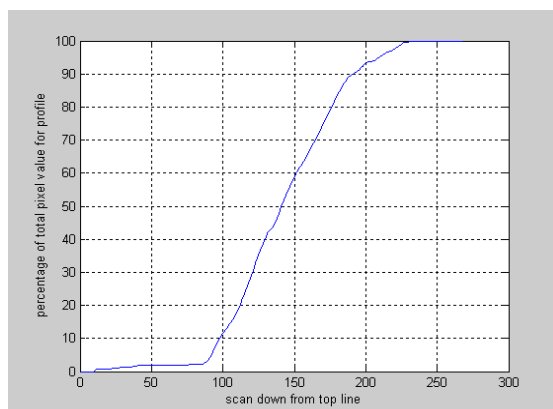


fig 2.29 Summed pixel values as a percentage of the total

The small percentage up to 90 in fig 2.29 is due to noise, which is not actually easily seen in fig 2.24. From then on the percentage rises steeply which corresponds to entering the systolic region. If the image has relatively low noise a threshold of 5% can be used to bring a profile check to a stop and thus locate point of maximum or minimum velocity.

If we now go back to a previous image as in fig 2.30 and use the percentile method we can see how it compares to the edge detection and ten-pixel-average methods. Two clicks are made again and all the profiles between the clicks checked for the point where a threshold of 20% or greater is reached. All these points are plotted in an envelope curve then smoothed – shown in fig 2.31. Note that this way if a 20% threshold is used this is equivalent to saying that 80% of the total pixel value is under the curve. This method is often used by researchers in this field but is influenced greatly by noise.

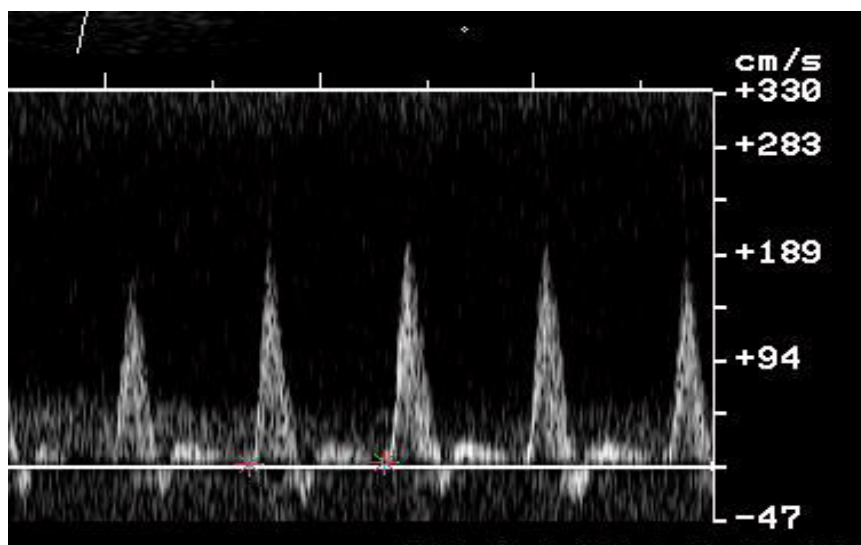


fig 2.30 Previously used image to try the percentile method on

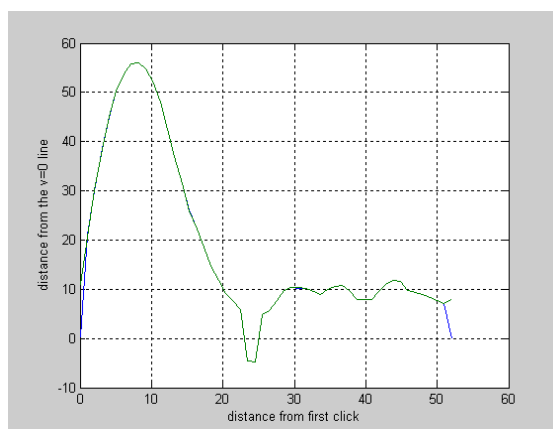


fig 2.31 Smoothed envelope curve for selected cycle in fig 2.30

Obvious noise in the image of fig 2.30 necessitated a threshold of 20% so the envelope is much lower compared to the edge detection method and the ten-pixel-average method. This method tends to be influenced by noise a great deal but is quick like the ten-pixel-average method. In a later chapter comparisons with the other four methods will be given with respect to average pixel value and ratio of areas. The edge method seems to take the longest time to arrive at maximum and minimum velocity envelopes.

CHAPTER THREE

Curve fitting and ratios

3.1 Least squares

Having created smoothed envelope curves using any of the four methods, it is now necessary to fit a best-fit curve over the data to enable Matlab to calculate areas (Or alternatively, simply sum up the velocities, pixel by pixel; this option not taken because Fourier series amplitudes will be needed for future ratios). For this task, using the Least Squares method, I will employ the Fourier series in equation (4) again. The number of components in the series can be chosen and the program work out the coefficients. Choosing a greater number of components makes the curve fit more tightly.

While three components in the Fourier series produce a systolic/diastolic curve, as in fig 3.1, it does not match the envelope very well. At this point the Levenberg-Marquardt ⁵ algorithm could be used to better position the three component-fit, or just simply use more Fourier components. If fifteen components are used as in fig 3.2, the fit hugs the data more closely. The way I have set up the code for least squares in the program results in the fit curve starting at zero (y axis) and finishing at zero so a large number of components is necessary to fit an envelope that does not finish or end at zero (see fig 2.24).

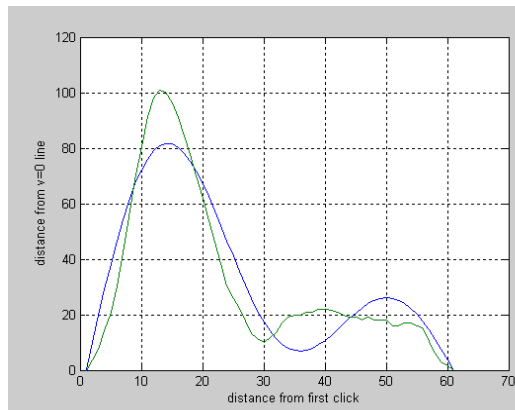


fig 3.1 Three component fit

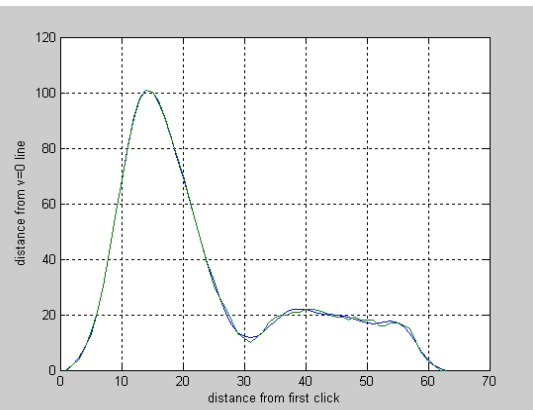


fig 3.2 Fifteen component fit

It is necessary to point out here that in some images reverse flow is observed. In other words some bright pixels (other than noise) are noticed below the zero velocity reference line as in fig 3.3. If it is necessary to capture this reverse flow in the final envelope curve then the code will have to work out from the maximum velocity envelope and the minimum velocity envelope when to change from one to the other. This is easily done by comparing absolute values of the data from both envelopes and choosing the greatest for each point. Again two clicks are made as in fig 3.3, the program works out the maximum and minimum velocity envelopes and then chooses the point from one or the other depending on which has the greater absolute value. Fig 3.5 is the result. This procedure can result in total envelope curve, which is distorted due to the lower curve capturing noise, giving it a bigger absolute value than the upper curve.

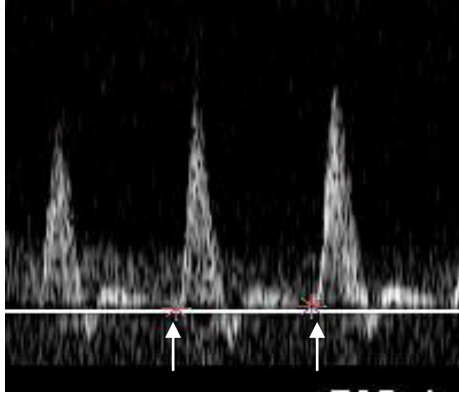


fig 3.3 Image showing reverse flow and clicks

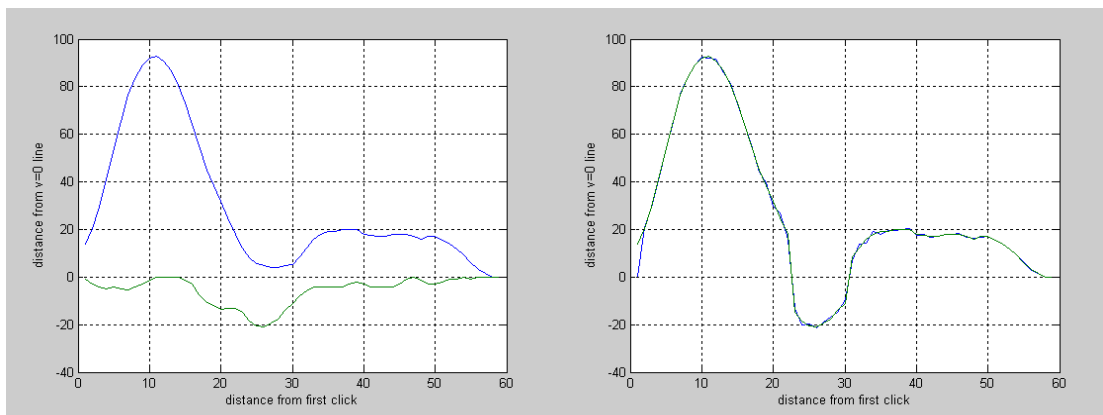


fig 3.4 The max. and min. vel. envelopes

fig 3.5 Merging both envelopes and fitted curve

3.2 Areas and ratios

Matlab can easily find the area of systole above the zero velocity line, then calculate the area of diastole, and then take the ratio:

$$R = \frac{\text{area systole}}{\text{area diastole}} \quad (5)$$

Not all images have reverse flow (which is included in diastole). Fig 3.6 shows an image with no reverse flow in most of its cycles.

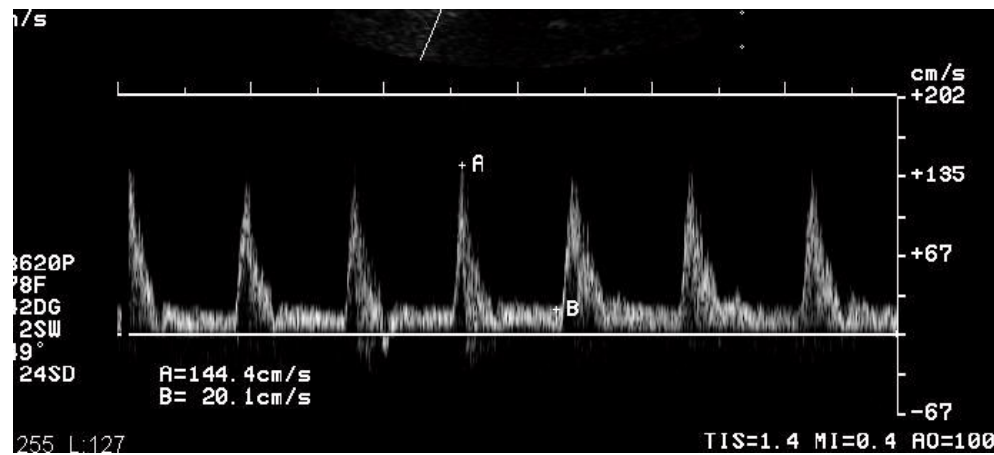


fig 3.6 Image with very little reverse flow

In order to find the point where systole ends and diastole begins in the envelope curve, either another click is necessary on the original image or the program first locates the systole peak then moves down the slope to the right and is made to stop once the curve starts rising again. Allowing the program to sort out itself the transition point has the advantage of not introducing more errors due to the positioning of clicks, but can meet with problems. If systole has a small valley in it some way down the slope the program will be stopped too early. Some images have this feature and it can depend on the threshold chosen. In a few cases diastole carries straight on from systole, continuing to fall at a slower rate so the program will not stop. Errors involved in using three clicks will be investigated in chapter 4. A much more time consuming process could involve looking for this point on the envelope curve (zooming if necessary) and keying in the number.

The ten-pixel-average method can be used on all complete cycles in fig 3.6 as an example. Smoothing, least squares, area calculations and ratios are also calculated. The average pixel threshold is 30 and three clicks were used. Table 3.1 shows the five area ratios, their mean value and standard deviation.

ratio1	ratio2	ratio3	ratio4	ratio5	image	ave	stdev
1.5654	1.4848	1.5448	1.6143	1.5535	1	1.552560	0.046447

table 3.1 Five area ratios, average value and standard deviation for the image in fig 3.6

The standard deviation gives some idea of the range of area ratios in an image (its significance limited because of small sample). The number of cycles in an image can vary and not necessarily all can be used. In the range of images in this study at least two cycles can be used.

For the number of images a person may have in their set, some may be scanned while lying horizontal and some after tilt. A simple plot of area ratio to image number may indicate when the person was tilted to 60° degrees from a supine position or more importantly when fainting occurred.

Some images have cycles with reverse flow in their diastolic region. Then equation (4) still holds except that the reverse flow will be subtracted from the diastolic region above the $v=0$ line:

$$R = \frac{\text{area systole}}{[\text{area diastolic (area above - area below)}]} \quad (6)$$

In some images the area of the diastolic area above the $v=0$ line is very small and the area of reverse flow larger, so the ratio R will be negative. It now becomes very important to get a good value for the first cross over point of the envelope curve over the zero velocity line as in fig 3.7 (near 23). *From then on the area calculation will automatically include the subtraction in Matlab.* Matlab can easily detect the crossover point.

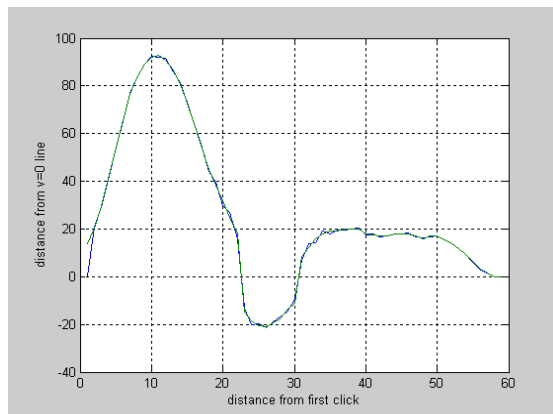


fig 3.7 Cross over point (near 23)

Another possibility is to add the area of reverse flow to the upper area of diastole so that the area ratio becomes:

$$R = \frac{\text{area systole}}{[\text{area diastole (area above + area below)}]} \quad (7)$$

This index will avoid negative values and be sensitive to changes in diastole. Two crossover points can easily be detected by Matlab and hence the three separate areas stored.

3.3 Loupas Index

Loupas⁶ suggested a ratio, which uses the amplitudes of the harmonics in the Fourier series used for the best fit of the maximum velocity envelope (note section 3.1).

$$\frac{\sum a_i^2}{V_{ave}^2} \quad (8)$$

a_i - harmonic amplitudes
 V_{ave} - average value of the maximum velocity envelope for diastole

The numerator will include the whole envelope rather than just systole. If the image cycles contain reverse flow in diastole, then the value of the ratio will depend on whether or not the absolute value diastolic velocities are averaged before squaring. This index could be compared to the other two for a sequence of images (see section 4.4).

3.4 Average pixel value

3.4.1 Noisy images

The average pixel value can be calculated for all the pixel spaces contained between the upper envelope and the zero velocity line for any cycle as in fig 3.8. Or if reverse flow is required, the average pixel value between the total curve and the zero velocity line as in fig 3.9. These curves change depending on the threshold chosen. A relationship will exist between threshold, area ratio and average pixel value.

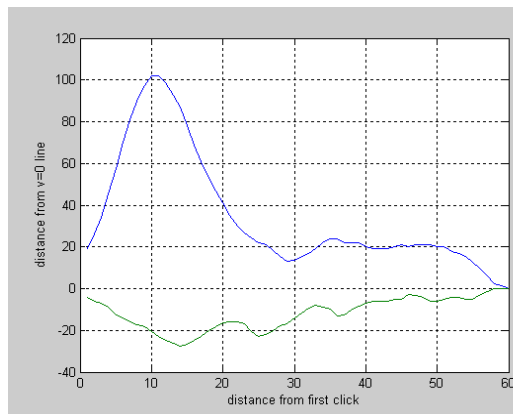


fig 3.8 Top curve used if reverse flow is not needed

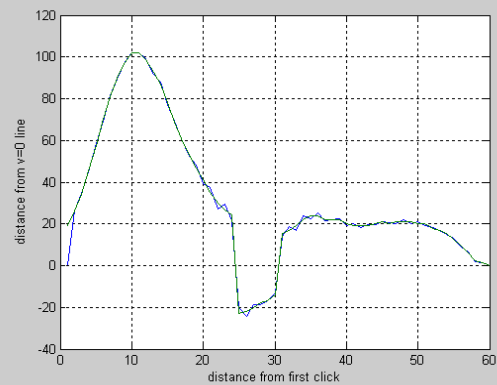


fig 3.9 Total curve including reverse flow

The four programs can now be modified to repeat the whole process for a range of thresholds. It takes some time for Matlab to calculate the average pixel value for each threshold. The ten-pixel-average method is the quickest also taking into account loops for threshold increments. So this version is now used to plot graphs of average pixel value versus threshold and area ratio versus threshold.

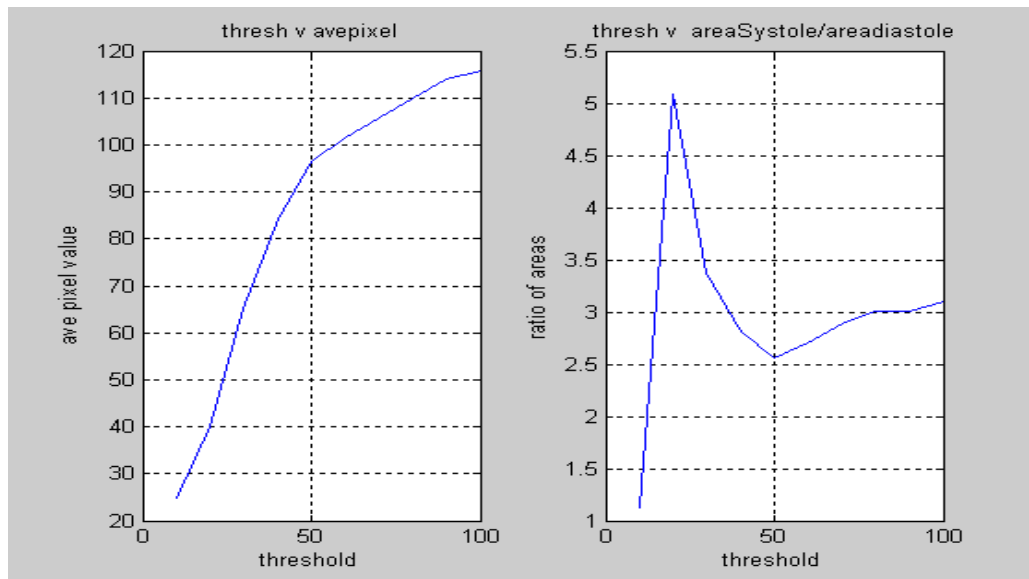


fig 3.10 Average pixel value vs threshold **fig 3.11** area ratio vs threshold

First we can choose a previous image with some noticeable noise as in fig 3.3. Three clicks are made. The program does ten loops, working out the average pixel values and area ratios for the thresholds 10 to 100 (in steps of 10). Figs 3.10 and 3.11 show the plots and table 3.2 shows the actual data for the ten thresholds.

threshold	10	20	30	40	50	60	70	80	90	100
ave pix val	24.7	40.0	65.2	84.1	96.4	101.4	105.7	110.0	114.0	115.90
area ratio	1.12	5.08	3.37	2.80	2.56	2.70	2.88	3.00	3.01	3.10

table 3.2 Average pixel values and area ratios for ten thresholds between 10 and 100

It is expected that the average pixel value will increase as the threshold increases due to the program missing the maximum speed and selecting a slower speed. Bearing in mind the scaling, at a threshold of 50, there is a change in the rate of increase of the average pixel value with threshold. A linear regression line put through the points with thresholds 10 – 50 has a gradient of 1.875 (correlation = 0.994) and a linear regression line put through points with thresholds 60 – 100 has a gradient of 0.399 (correlation = 0.993). At a threshold of 50 the program has overcome the background noise.

In the area ratio plot (fig 3.11), at a threshold of 50 the area ratio stops varying wildly and settles to a smaller change – 2.5 to 3.0. In fact this behaviour is quite characteristic when noise is present and accompanies an important change in the maximum velocity envelope; an *acceptable* envelope as outlined in the next chapter.

3.4.2 Images with little noise

If we use the program on a cycle in an image such as fig 3.6 (little obvious noise), the slope change of the average pixel curve is not so sudden and the area ratios show a smaller variation across the thresholds (1.5 – 1.62) than the previous image across the thresholds – see figs 3.12 and 3.13. The actual average pixel value and area ratio data is displayed in table 3.3.

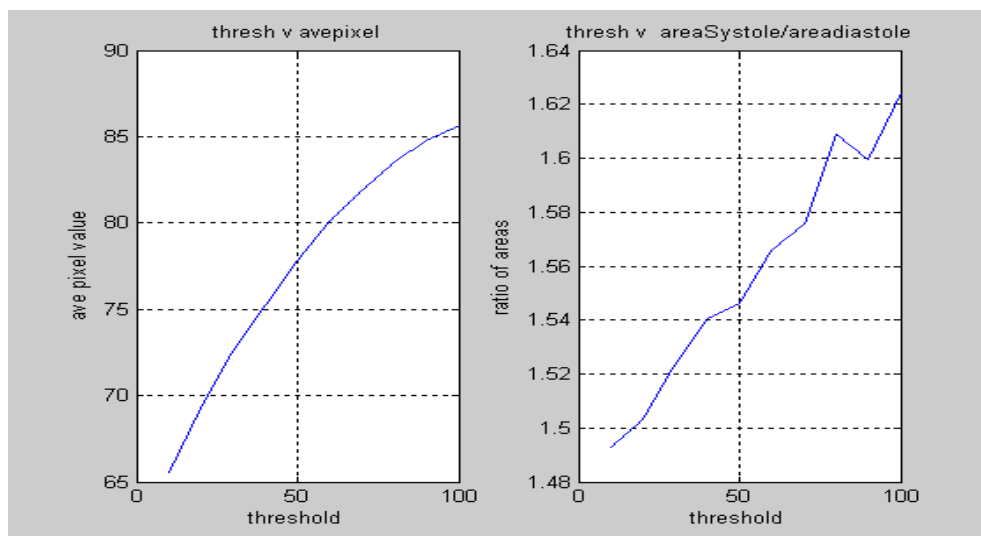


fig 3.12 Average pixel value vs threshold **fig 3.13** area ratio vs threshold

threshold	10	20	30	40	50	60	70	80	90	100
ave pix value	65.5	69.2	72.5	75.2	77.8	80.0	81.9	83.5	84.7	85.6
area ratio	1.49	1.50	1.52	1.54	1.54	1.56	1.57	1.60	1.59	1.62

table 3.3 Average pixel value and area ratio for ten threshold from 10 to 100 for figs 3.12 and 3.13

3.4.3 Examining more than one cycle in an image

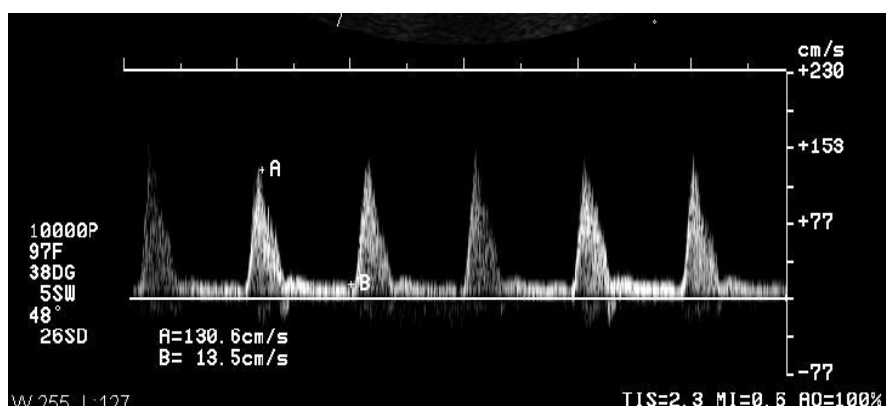


fig 3.14 Image with cycles that vary in pixel brightness

Fig 3.14 shows an image with five complete cycles. Two of the cycles have dimmer systole and one with a weaker diastole. If the procedure of finding average pixel values and area ratios for incremental changes in threshold is now applied to these five cycles the graphs in figs 3.15 and 3.16 are obtained.

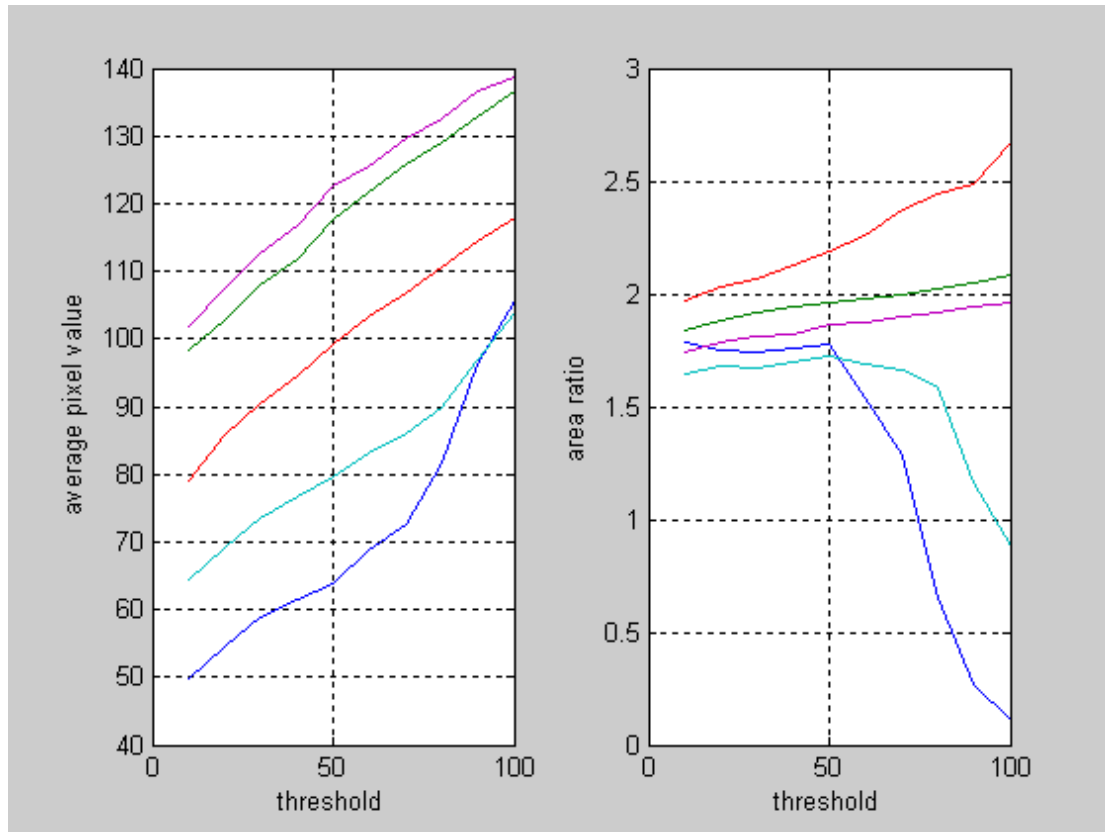


fig 3.15 Average pixel value vs threshold
for the five cycles in fig 3.14

fig 3.16 area ratio vs threshold
for the five cycles in fig 3.14

Of particular interest is the area ratio versus threshold graph in fig 3.16. The line that drops the down the most (to 0.1) belongs to cycle 1 from the left in fig 3.14. It has a dim systole and so as the threshold increases the program will select a slower than maximum speed in systole than in diastole reducing the ratio R . The next line that drops (to 0.9), belongs to cycle 4 from the left, and drops for the same reason as cycle 1. Cycle 3 has a slightly weaker diastole (or some small gaps in it) which means as the threshold increases the program will select a slower than maximum speed in diastole than systole so the ratio R will increase. The top line in the area ratio graph of fig 3.16 shows this (reaching 2.7). The remaining two cycles have equal brightness systole and diastole, and the area ratios do not vary as much over the incremental threshold change.

Thresh	10	20	30	40	50	60	70	80	90	100
Mean R	1.79	1.82	1.84	1.86	1.90	1.86	1.84	1.72	1.58	1.54
St dev	0.12	0.14	0.15	0.17	0.18	0.28	0.40	0.67	0.87	1.03

table 3.4 Mean area ratios and standard deviation for 5 cycles as threshold changes

When examining the range of images belonging to a particular person; using this version of the program, a threshold should be chosen to deal with the noise in the worst image and kept constant and in each image choosing enough cycles to average but trying to avoid unequal brightness in systole and diastole. It is obviously not possible to carry out an average pixel versus threshold plot for all cycles in all images in the fainting study!

·
·

CHAPTER FOUR

Errors and Comparisons

4.1 Clicking errors

Errors are obviously introduced into the process by clicking on the image, even when the image is expanded to full screen. Fig 4.1 shows an image with four cycles that can be processed using one of the four programs and a mean value obtained for the area ratio. The last cycle on the right has a diminishing diastolic region, which may indicate that only three cycles should be used in this image. Fig 4.2 shows a zoomed picture of cycle one and the three crosses show the approximate positions (x coordinate) of the clicks. Each click could land in a range of two to three pixel columns. A number of clicking trials using one of the programs (ten-pixel-average method) will indicate the range of area ratio values

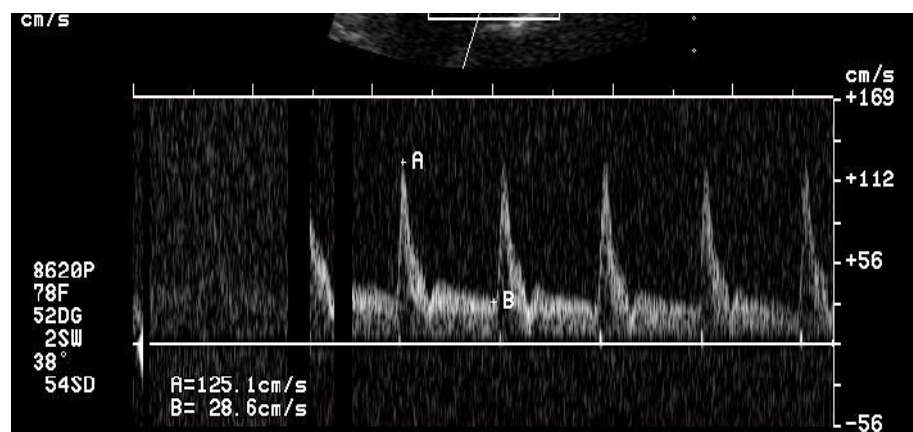


fig 4.1 Four cycles in this image can be used to find a mean value for the area ratio

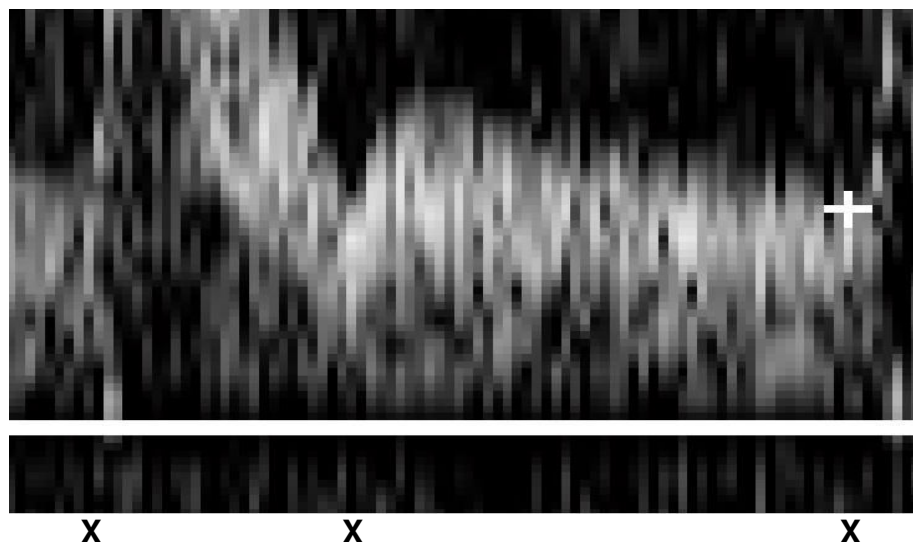


fig 4.2 Zoomed first cycle in fig 4.1 and approximate positions of the three clicks

Table 4.2 shows the results of 20 trials, clicking on each of the four cycles three times. The last cycle has a dim diastole toward the end so it is expected that its area ratio will be little higher.

trials	Cycle1	Cycle2	Cycle3	Cycle4	ave	std
1	0.9378	0.9402	1.1525	1.2304	1.0652250	0.1491845
2	0.9353	0.9983	0.9993	1.3305	1.0658500	0.1789551
3	0.9555	0.9922	1.0166	1.2412	1.0513750	0.1290175
4	0.9026	0.9824	1.0898	1.1833	1.0395250	0.1227607
5	0.9378	0.9922	1.0166	1.2125	1.0397750	0.1197682
6	0.9555	0.9402	0.9514	1.1145	0.9904000	0.0829857
7	0.9230	0.9824	1.1452	1.2125	1.0657750	0.1356130
8	0.9883	0.9649	1.0536	1.1661	1.0432250	0.0901064
9	0.9707	0.9488	1.0050	1.1513	1.0189500	0.0912144
10	0.9555	0.9136	1.0166	1.1661	1.0129500	0.1105127
11	0.9378	0.8949	1.0510	1.2229	1.0266500	0.1464688
12	0.9208	0.9649	1.0170	1.1145	1.0043000	0.0833265
13	0.9230	0.9649	0.9993	1.1971	1.0210750	0.1214266
14	0.9603	0.9157	0.9194	1.2577	1.0132750	0.1641984
15	0.9230	0.9289	0.9993	1.1145	0.9914250	0.0890708
16	0.9026	0.9757	0.9993	1.2718	1.0373500	0.1616301
17	0.9555	0.9560	0.9194	1.2304	1.0153250	0.1444038
18	0.9080	0.9136	0.9814	1.2701	1.0182750	0.1711656
19	0.9026	0.9457	0.9689	1.2412	1.0146000	0.1535440
20	0.8568	0.9289	1.0226	1.2071	1.0038500	0.1515512

table 4.2 Twenty trials of clicking in three places on each of the four cycles

These area ratios over the twenty trials are plotted in fig 4.3. As expected the fourth cycle has higher values for the twenty trials. Fig 4.4 shows the mean values of four cycles for each trial (upper curve) and the mean values of three cycles for each trial (if the cycle on the right is excluded).

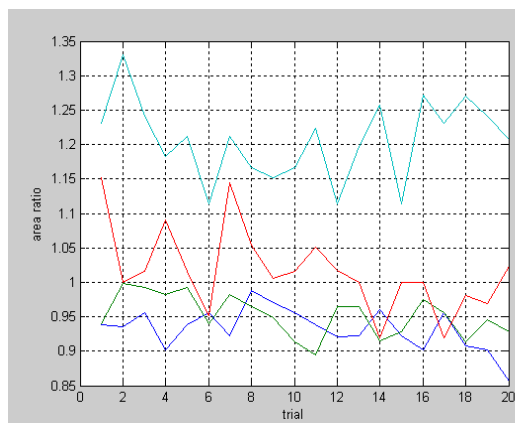


fig 4.3 twenty trial area ratios for four cycles in table 4.2

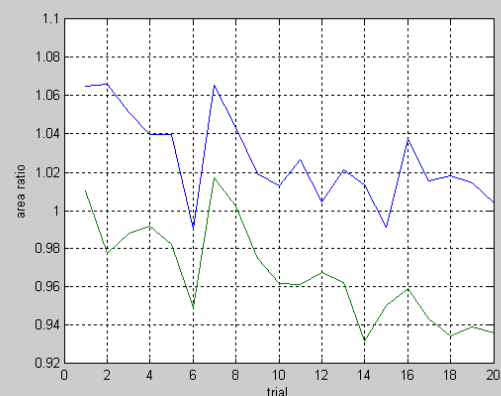


fig 4.4 Top line means for four cycles
Bottom line means for three cycles

The overall mean of the area ratios for the four cycles and the first three cycles are shown in table 4.3 The difference between these means is only 0.06 – *choosing cycles with bright systole and diastole is in general, preferable.*

mean1 = 1.0270	mean2 = 0.9670
range1 = 0.0754	range2 = 0.0851

table 4.3 Mean and range for four cycles,

Mean and range for three cycles

Of interest also is the position of the third click between systole and diastole. Four example, if we look at one of the clicking trials in cycle one (an area ratio of 0.9555 - occurs four times in table 4.2), The 'ca' code in Matlab shows that the three clicks on the image have x coordinates of $ca(1) = 307$, $ca(2) = 388$ and the click between systole and diastole $ca(3) = 337$. The difference $ca(3) - ca(1)$ should correspond to the 'minimum' position (between systole and diastole) on the envelope curve in fig 4.5, ($337 - 307 = 30$). The zoomed envelope curve in fig 4.6 shows the actual minimum value to have an x coordinate of 29; hence contributing to the area ratio errors.

A comparison can now be made between the mean values obtained using three clicks on an image and only two clicks on an image. After two clicks Matlab computes the smoothed envelope then a best-fit curve and then locates the minimum point between systole and diastole itself rather than another click being used. This method was mentioned in section 3.2. The peak value on systole is found then the program scans down the right side of the systole curve until the point at which it begins to turn upwards in diastole. This method could locate the wrong position if systole has a local valley before the point where systole meets diastole.

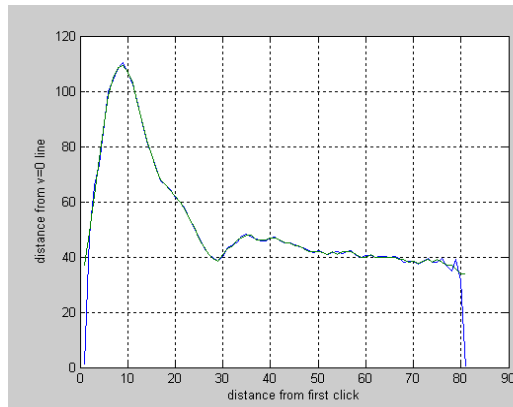


fig 4.5 Envelope curve for cycle one R=0.9555

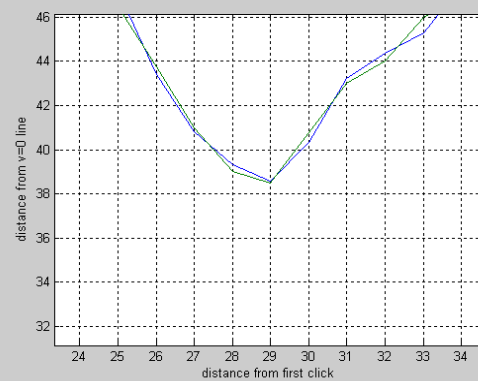


fig 4.6 Zoomed envelope curve

Table 4.4 shows the twenty trials using the cycles in fig 4.1 with only two clicks. Again graphs and mean values are obtained for both four cycles and three cycles (eliminating the cycle with diminishing diastole) - figs 4.7 and 4.8.

Table 4.5 shows the overall mean value for four cycles and the mean value for three cycles if we choose not to use the last cycle.

trial	cycle1	cycle2	cycle3	cycle4	ave	std
1	0.9378	0.9560	1.1042	1.1988	1.0492	0.1245008
2	0.9707	0.9457	1.0194	1.1833	1.0298	0.106827
3	0.9378	0.9457	1.0194	1.2903	1.0483	0.1654652
4	0.9378	0.9457	1.0194	1.3070	1.0525	0.1736166
5	0.8886	0.9457	0.9857	1.2464	1.0166	0.1582968
6	0.9378	0.9560	0.9857	1.1833	1.0157	0.113464
7	0.9378	0.9289	1.1042	1.2464	1.0543	0.1513163
8	0.9883	0.9457	1.0194	1.2464	1.0500	0.1344057
9	0.9378	0.9289	0.9857	1.2464	1.0247	0.1498901
10	0.9378	0.9983	1.0166	1.1833	1.0340	0.1050749
11	0.9883	0.9457	0.9857	1.2464	1.0415	0.137968
12	0.8886	0.9289	1.1042	1.2304	1.0380	0.1587696
13	0.9378	0.9983	1.0194	1.1833	1.0347	0.1049295
14	0.9547	0.9457	1.0194	1.2464	1.0416	0.1404567
15	0.9547	0.9457	1.0194	1.2464	1.0416	0.1404567
16	0.9883	0.9457	0.9857	1.2071	1.0317	0.1185478
17	0.9883	0.9289	1.0194	1.2464	1.0458	0.1389354
18	0.9378	0.9560	1.0194	1.2464	1.0399	0.14204
19	0.9547	0.9983	1.1042	1.2464	1.0759	0.1298491
20	0.9378	0.9457	1.0194	1.2071	1.0275	0.1252452

table 4.4 Twenty trials for clicking twice on the four cycles in fig 4.1

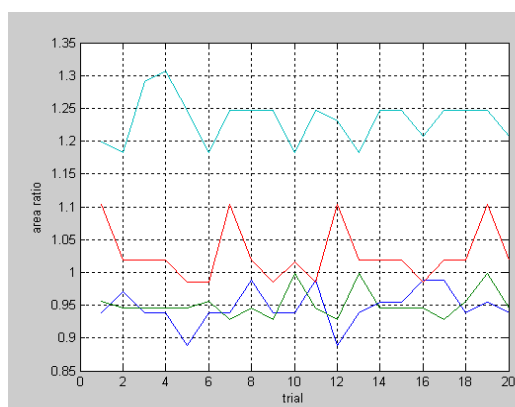


fig 4.7 Area ratios for the four cycles in fig 4.1 across twenty trials

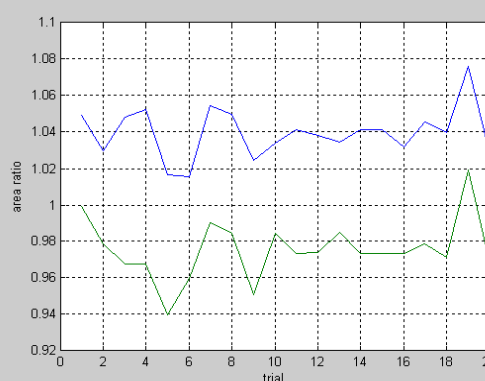


fig 4.8 Mean values for four cycles and three cycles (lower line)

mean1 = 1.0397	mean2 = 0.9756
range1 = 0.0602	range2 = 0.0791

table 4.5 Overall mean for four cycles

Overall mean for three cycles

Comparing table 4.5 with table 4.3 – mean1 (four cycles) and mean2 (three cycles are not significantly different. This is not to say that all images in the study will produce results as close as this but tends to suggest that if the three click mode is chosen it probably will not be too far from using two clicks. The reason for choosing three clicks is that if local valleys appear in systole as in fig 4.9, the third click will be very

near 35 instead of 24. If Matlab finds the third coordinate automatically it will have found 24.

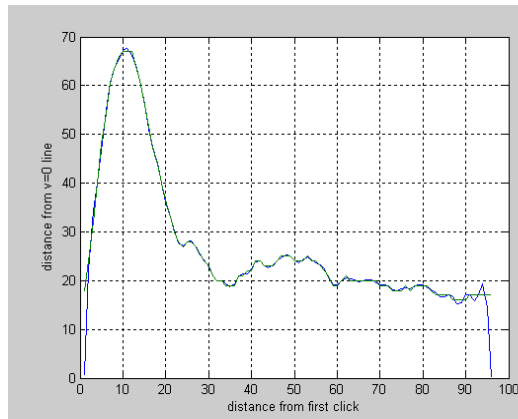


fig 4.9 Envelope with a local valley in systole

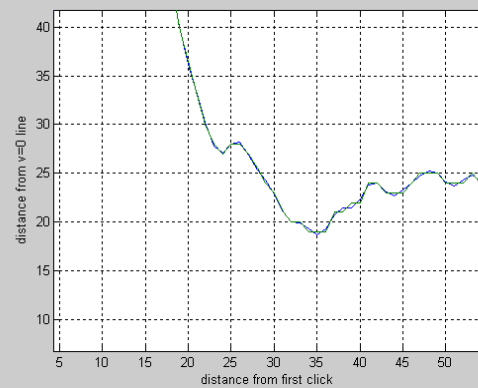


fig 4.10 Zoomed version of fig 4.9

There is a very rare occurrence in the best fit curve. The best-fit curve from which the areas are calculated may have a small upward portion in systole due to randomness as in fig 4.12. This gives the third coordinate in this example a value of 20 instead of 37 so the area ratio will be smaller than it should be (0.7426 instead of 1.0194). So in a sequence of images and cycles if a value for R (area ratio) is smaller than expected a zoomed picture of the envelope curve should be inspected for a sudden rise in systole instead of at the boundary of systole and diastole.

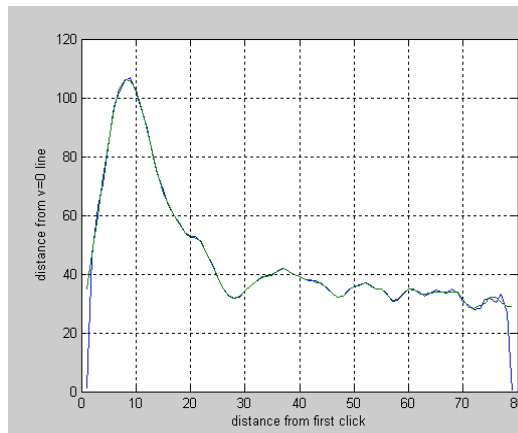


fig 4.11 Envelope with a bent systole but no local valley

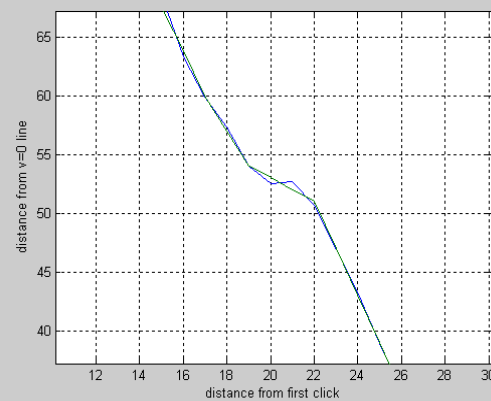


fig 4.12 The best fit curve has an upward turn, giving an early third position

To complete the analysis on all images in the clinical study, clicking three times on an image cycle is the better choice if there is no reverse flow. If reverse flow is present, Matlab can detect crossover points ($v=0$ line) automatically and the three separate areas found and stored. For images with a large number of cycles two or three cycles may have to be evaluated together⁵ to avoid clicking errors.

4.2 Comparing the four programs on one image

The four programs (ten-pixel-average, edge filter, differential and percentile) can now be compared. For the first comparison, each program can be used on the image used in section 4.1 (fig 4.1).

4.2.1 Ten-pixel-average method

Three clicks are made on cycle *one* in fig 4.1. The ten-pixel-average program is now looped ten times to produce smoothed envelope curves for the thresholds – 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Figs 4.13 and 4.14 show the set of graphs for these thresholds. From a threshold of 40 onwards the maximum velocity envelope becomes *acceptable* and fig 4.15 (and table 4.6) show the near constant area ratio from this threshold onward. Note the characteristic change in the average pixel value versus threshold in the first graph of fig 4.14 at a threshold of 40 in this case. For thresholds greater than 100 the program selects velocities slower than the maximum in the systolic and diastolic regions and the area ratio diminishes; the envelope will become distorted. The wide range of thresholds; 40 to 100, for acceptable envelopes and near constant area ratio in this version of the program is fortunate for noisy image sets.

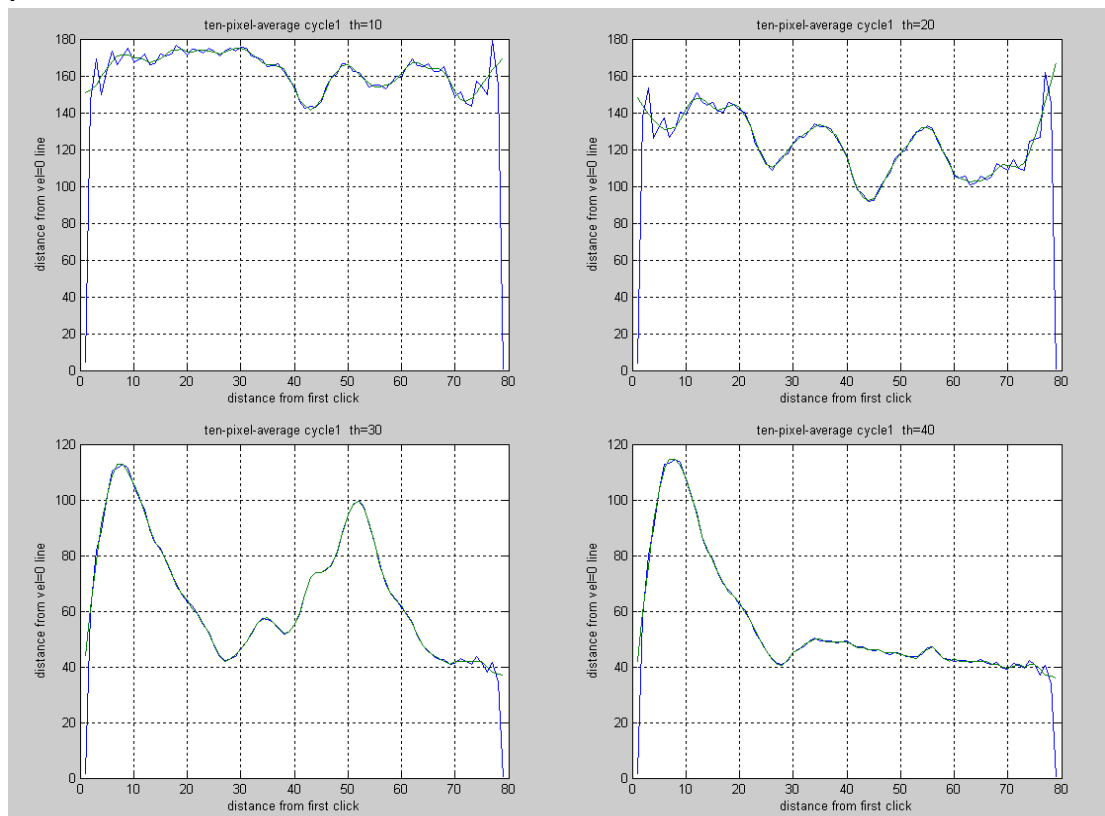


fig 4.13 Envelope/best-fit curves for thresholds 10 to 40 in steps of 10 for cycle one in fig 4.1

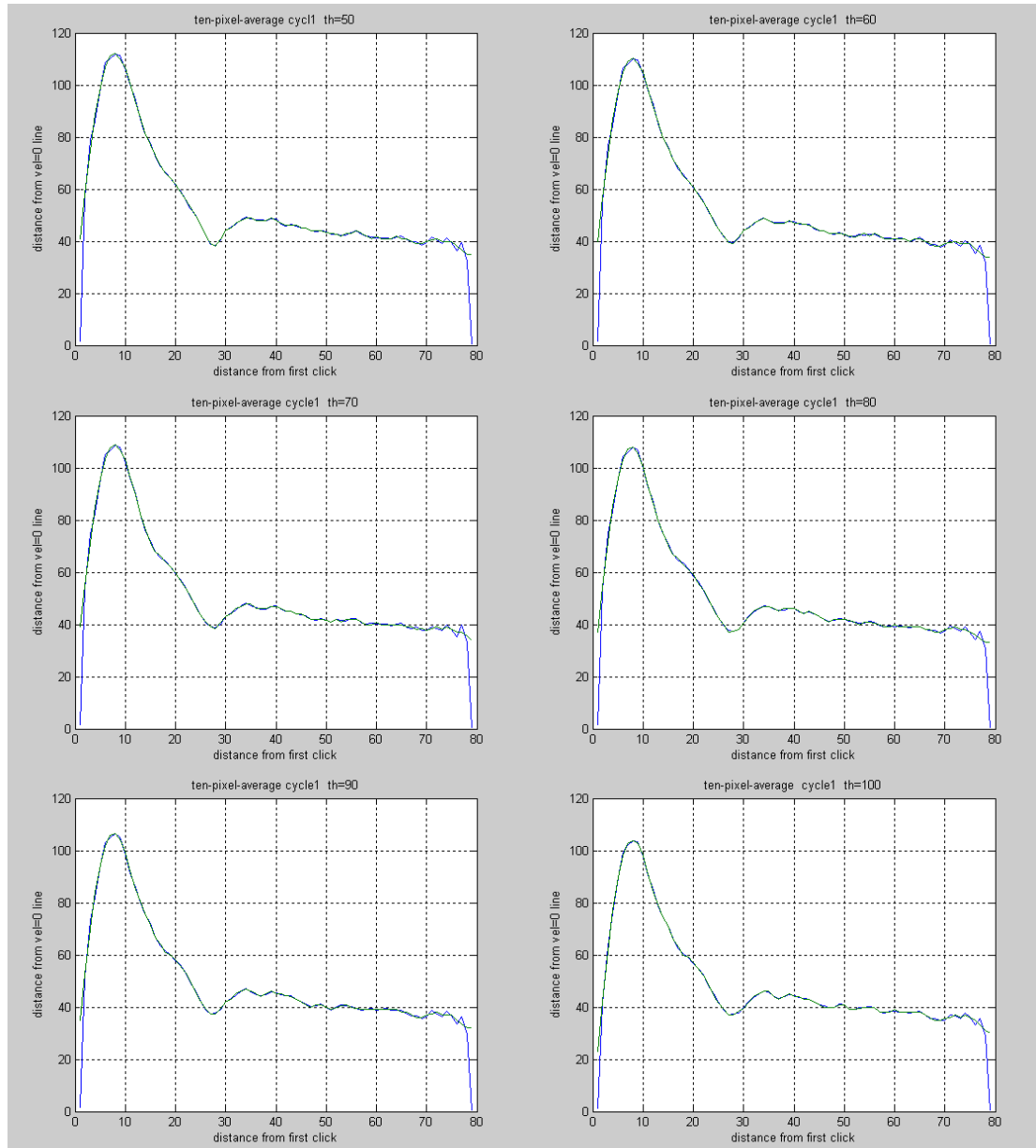


fig 4.14 Envelope/best-fit curves for thresholds 50 to 100 in steps of 10 for cycle one in fig 4.1

The peak values of systole and the peak values of diastole can be plotted against threshold (thresholds 40 to 100 give acceptable envelopes) to indicate the reduction in height as threshold increases. Fig 4.16 shows the two graphs for this and linear regression lines drawn in. Both regression lines have a correlation of 0.988 with the systolic peak line with gradient -0.160 and the diastolic line gradient -0.067 . The systolic peak is diminishing at a slightly faster rate. Using the command 'pixval' in Matlab, the height of the systolic peak for cycle one is 122 and the pixel brightness 55.

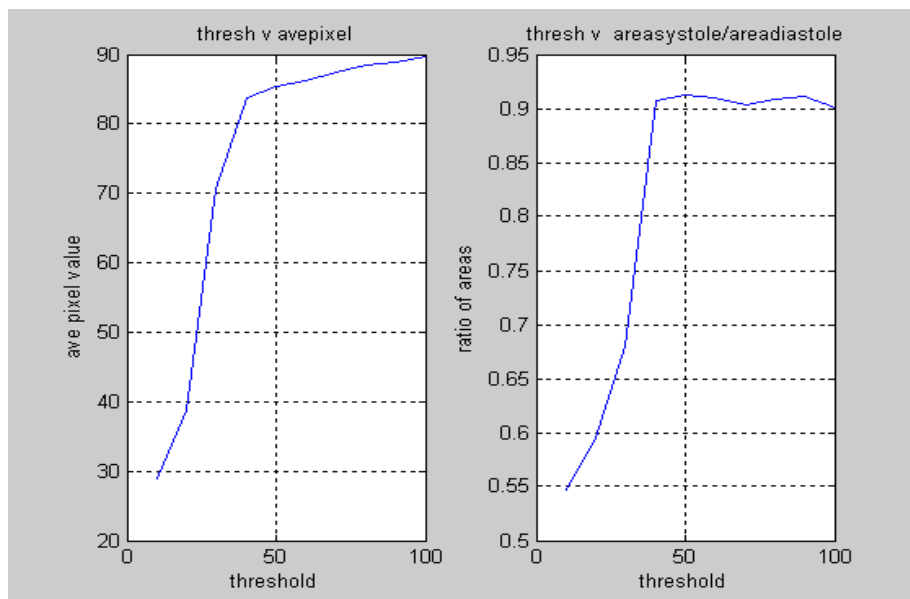


fig 4.15 Average pixel vs threshold and area ratio vs threshold for cycle one in fig 4.1

th	10	20	30	40	50	60	70	80	90	100
avepx	28.9	38.8	70.6	83.8	85.4	86.3	87.4	88.4	88.9	89.8
areaR	0.55	0.59	0.68	0.91	0.91	0.91	0.90	0.91	0.91	0.90

table 4.6 Data for fig 4.14

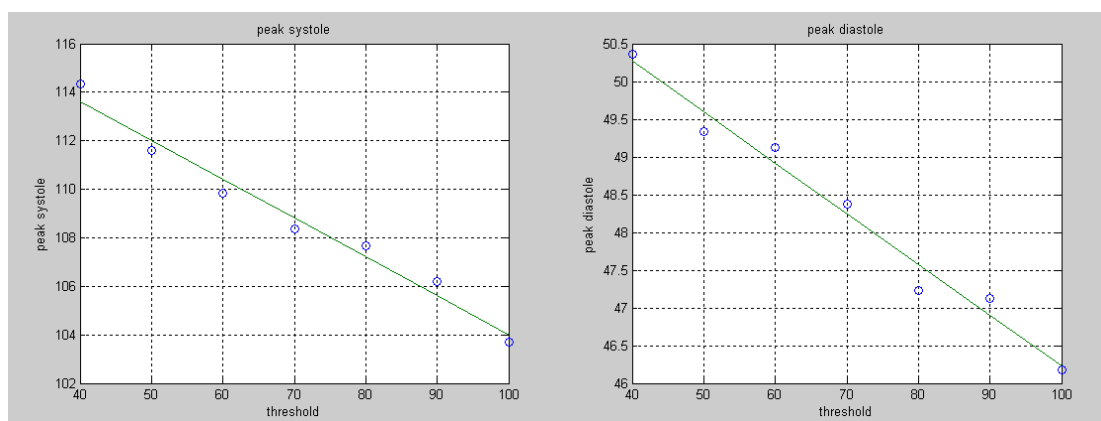


fig 4.16 Changes in systolic and diastolic peak values for thresholds 40 to 100 using the ten-pixel-average method

The looped ten-pixel-average program can now be applied to the other three cycles on the image in fig 4.1. The graphs in fig 4.17 display area ratio and average pixel value versus threshold for the four cycles. As already noted in section 4.1, the forth cycle has a weakening diastolic portion which tends to cause the area ratio to increase more significantly as threshold increases.

It is preferable to keep the threshold constant when evaluating a particular person's image set. However if background noise in any particular image is significantly different from the rest, the data in the results just obtained show that a small adjustment in threshold to overcome the noise will not give an inconsistent result in area ratio.

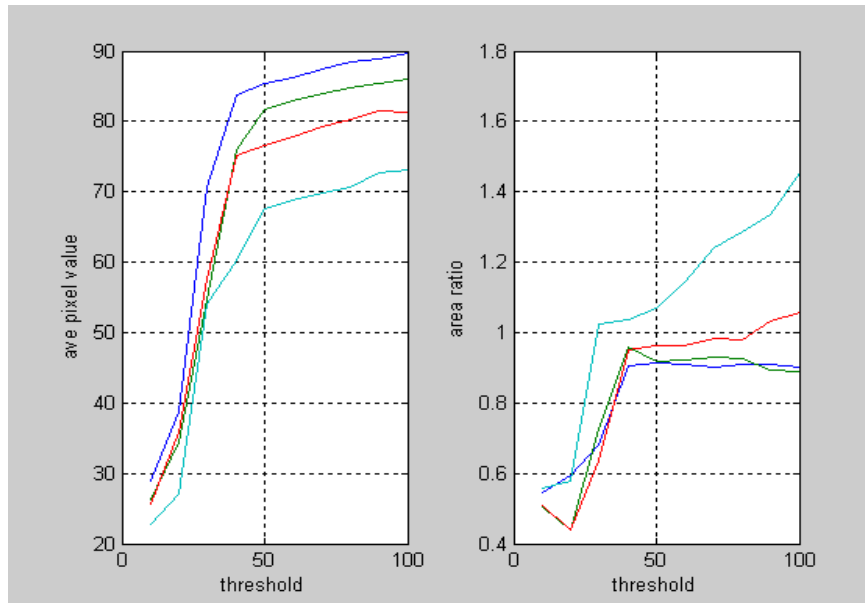


fig 4.17 Summary of ave. pixel value and area ratio vs threshold for four cycles in fig 4.1 using the ten-pixel-average program

threshold	10	20	30	40	50	60	70	80	90	100
R	0.53	0.51	0.77	0.96	0.97	0.99	1.01	1.03	1.04	1.07

Table 4.7 Mean area ratios for the four cycles as threshold increases

4.2.2 Edge detection method

Three clicks are made on cycle one in fig 4.1. The edge detection program is now looped six times to produce smoothed envelope curves for Gaussian mask thresholds of 30, 40, 50, 60, 70, 80 (see section 2.3). Fig 4.18 shows the actual edge images for thresholds 30 to 80. The same three click locations are used for all thresholds.

As the threshold increases more noise is left in the image however the cycles have more complete edges. A threshold of 50 enables three well-formed cycles to be used with no surrounding noise.

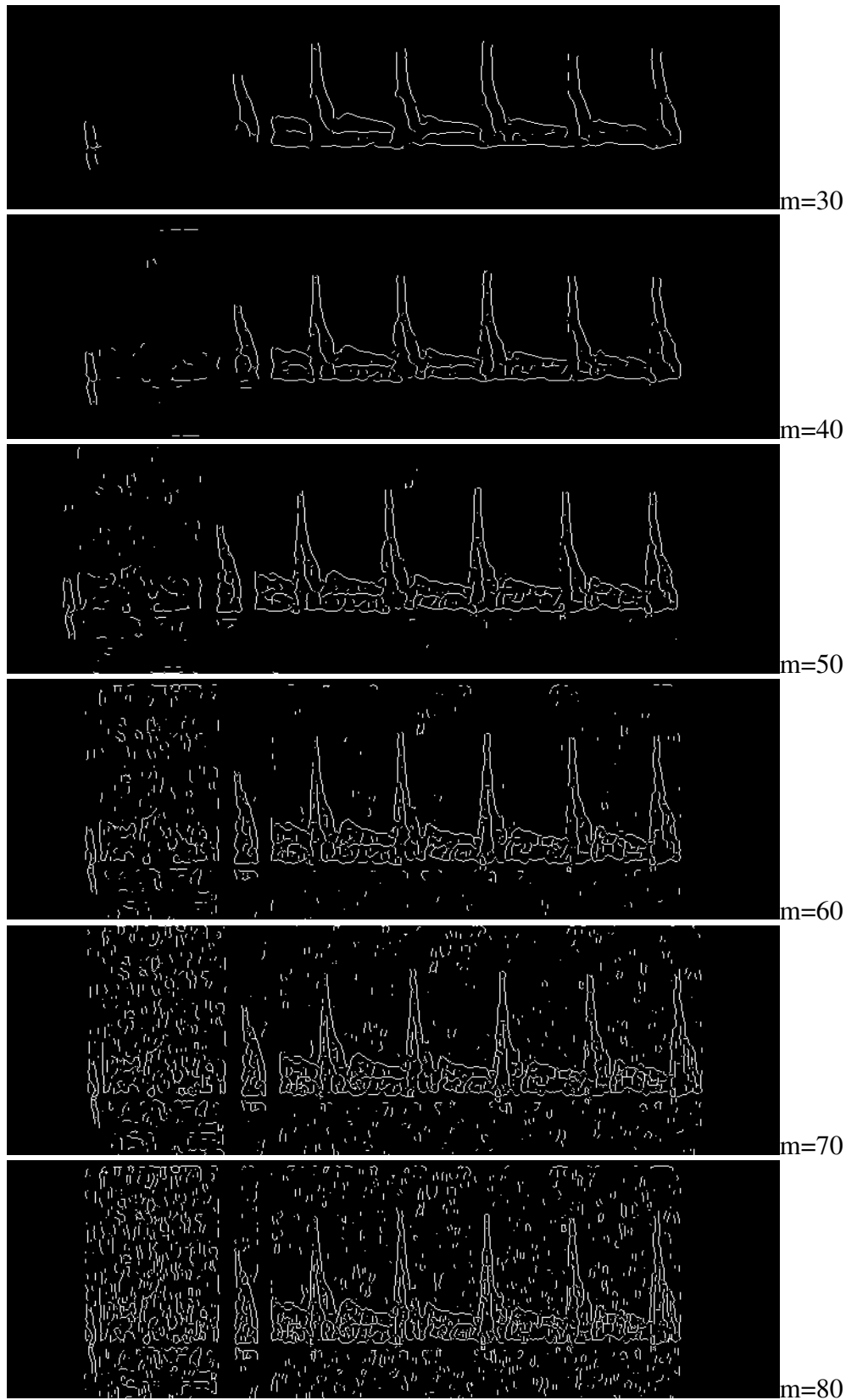


fig 4.18 Edge detection of cycle one in fig 4.1. Thresholds 30 to 80 in steps of 10

The six smoothed envelope curves for cycle one, as the Gaussian threshold increases, are shown in figs 4.19. Thresholds 30, 40 and 50 in fact produce well-defined systolic and diastolic cycles. Even the absence of the top edge of the systole for threshold 30

has not distorted the envelope curve for that threshold (smoothing). More noise is captured as threshold (m) is increased.

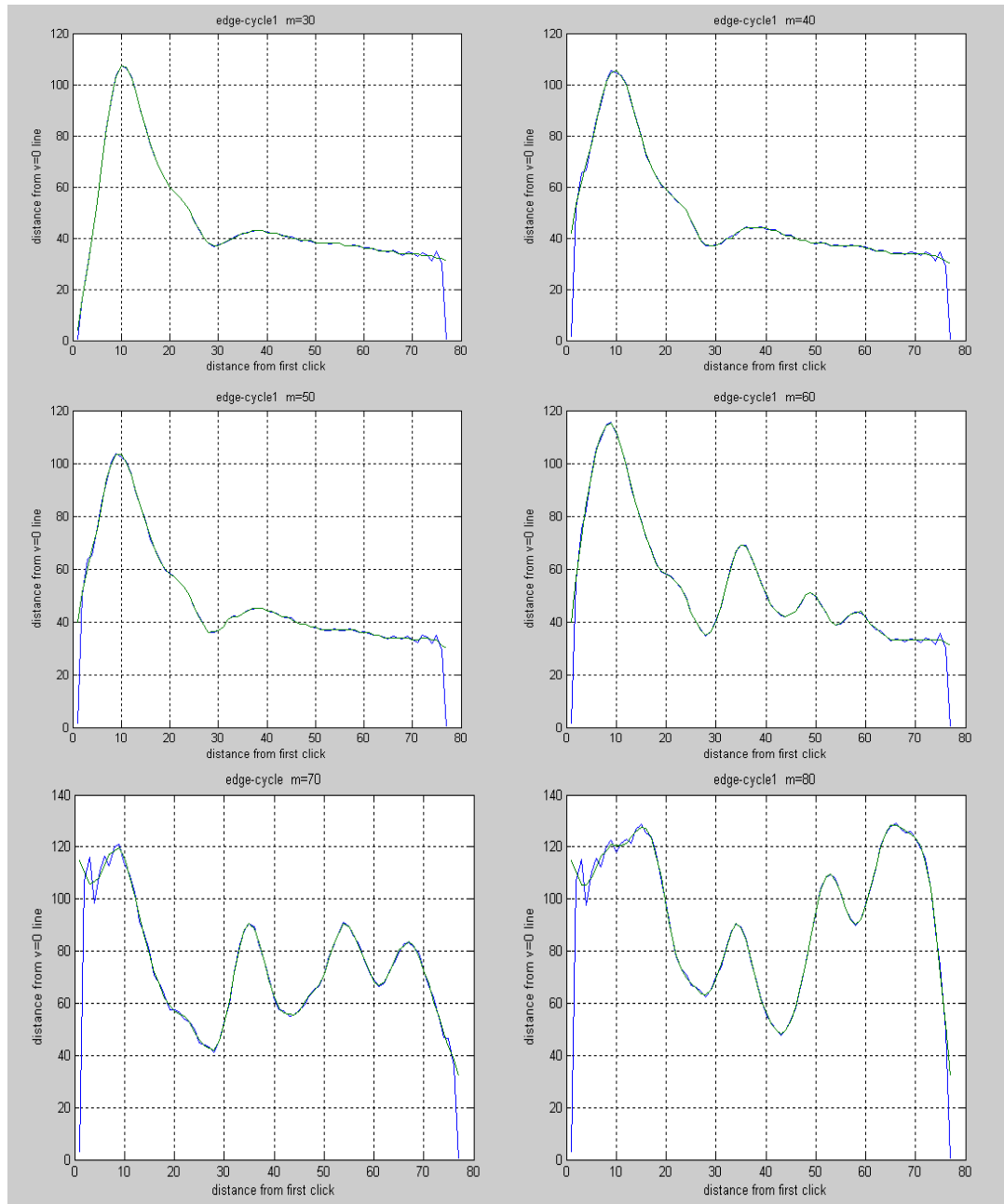


fig 4.19 Smoothed envelopes for edge detection method of cycle one in fig 4.1
Thresholds 30 to 80 in steps of 10

The area ratio for each of these thresholds, are shown in table 4.8 with the first three being significant, and from then on the area ratio decreasing. Fig 4.20 shows the graphs of area ratio and average pixel values under the curves as threshold increases.

threshold	30	40	50	60	70	80
R	0.985	1.049	1.028	0.952	0.650	0.624

table 4.8 Area ratio versus Gaussian threshold for cycle one in fig 4.1

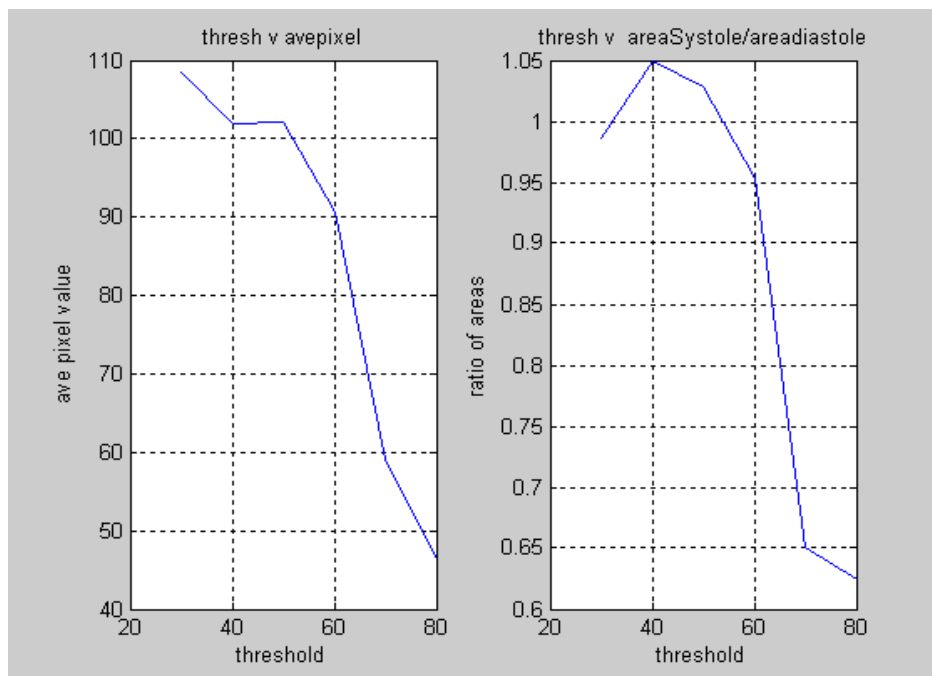


fig 4.20 Area ratio and average pixel value vs threshold for cycle 1 - edge method

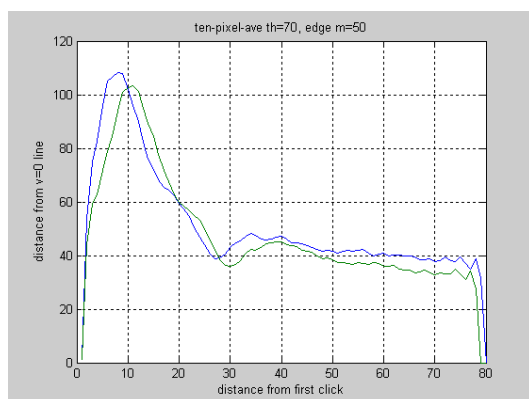


fig 4.20a Maximum velocity envelopes for the ten-pixel-average (th=70) and edge (m=50) graphed together

For comparison the maximum velocity envelopes for the ten-pixel average (threshold = 70) and the edge (m=50) methods are graphed together in fig 4.20a.

When the whole four cycles are evaluated using the looped edge method, the area ratios can be averaged for each threshold and they are shown in table 4.9. Fig 4.21 shows the graph of the area ratio versus threshold for all four cycles in fig 4.1 using the edge method.

threshold	30	40	50	60	70	80
R-ave	0.9760	1.0954	1.1766	1.0295	0.6642	0.5914

table 4.9 Average area ratio vs threshold for the four cycles in fig 4.1 - edge method

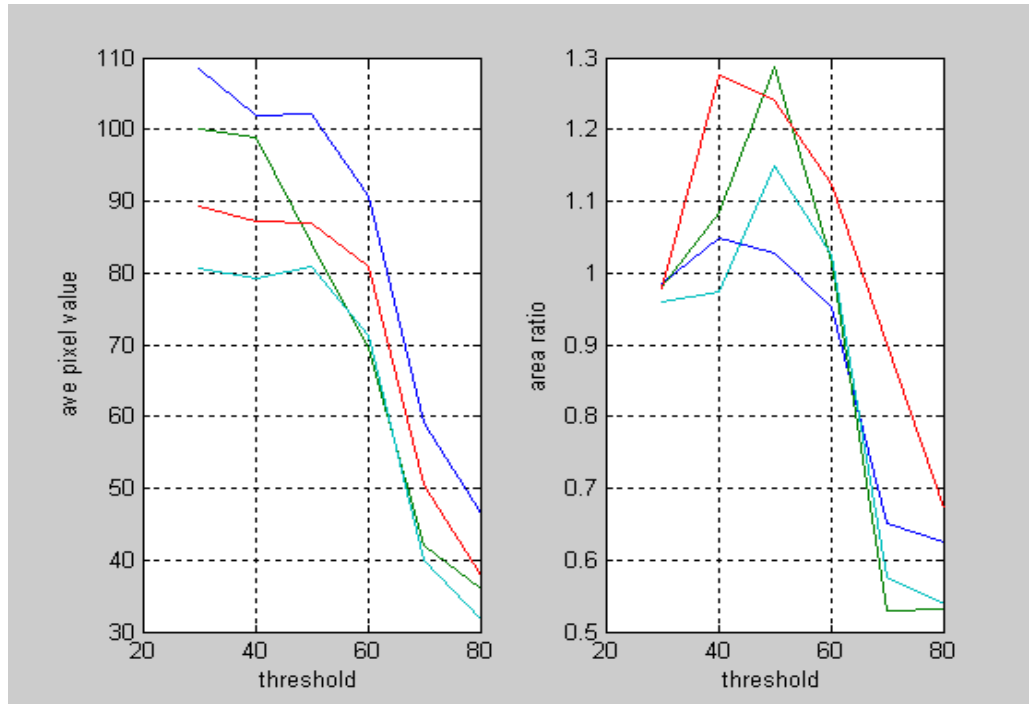


fig 4.21 Area ratio and average pixel value vs threshold for all four cycles in fig 4.1 – edge method.

4.2.3 Differential method

Three clicks are made on cycle *one* in fig 4.1. The differential program is now looped seven times to produce smoothed envelope curves for the thresholds – 500, 1000, 1500, 2000, 2500, 3000 and 3500. Fig 4.22 shows the set of graphs for these thresholds. An acceptable envelope curve is found for thresholds near 1500 and not usable for thresholds either side. As seen in section 2.4, profile scans for the differential method show noise peaks varying considerably, so thresholds become a matter of hundreds when changed in the differential method.

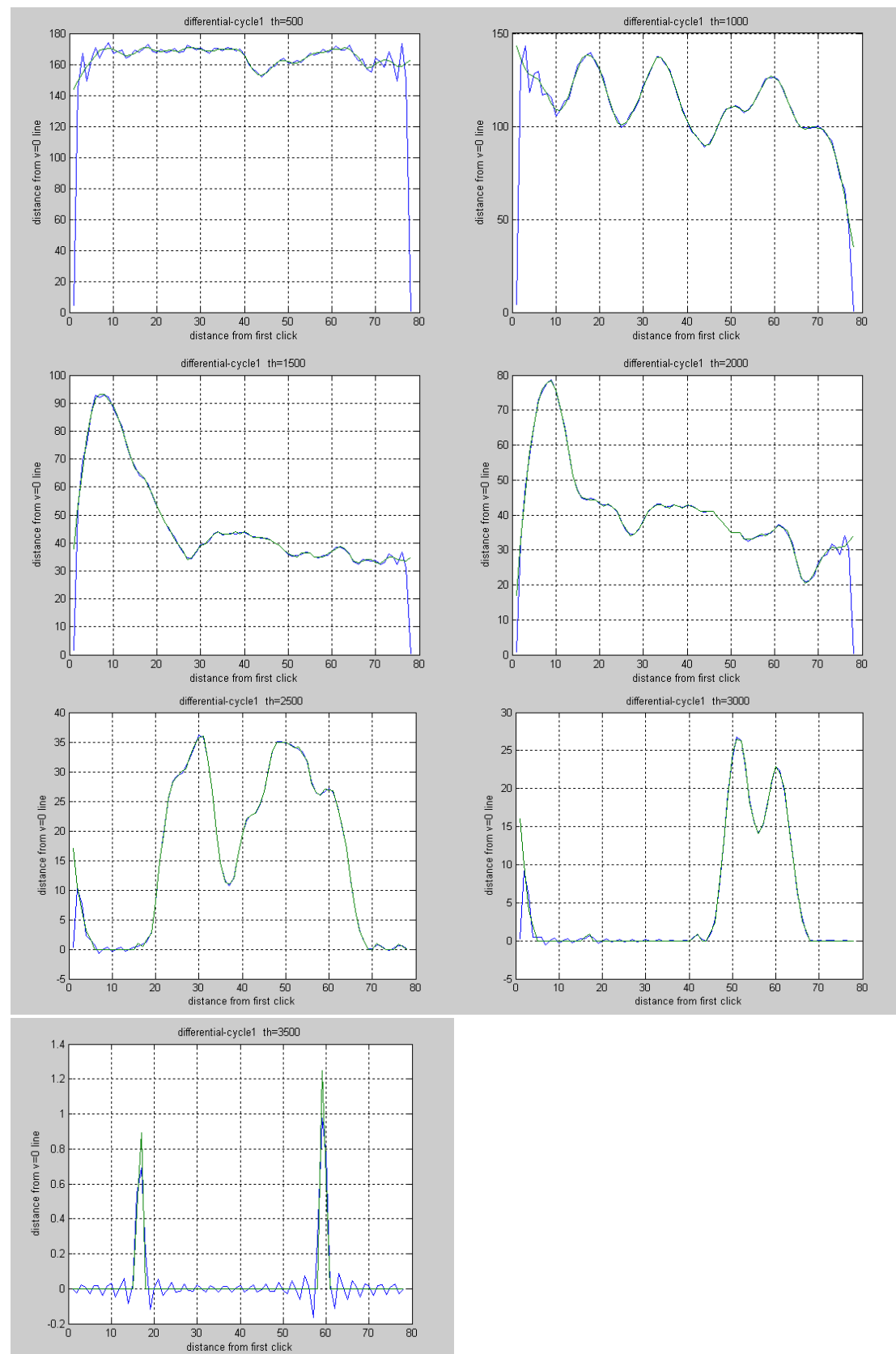


fig 4.22 Smoothed envelope curves for cycle one in fig 4.1 using the differential method

Fig 4.23 shows the change of area ratio and average pixel value with increasing threshold. The actual data for area ratio is shown in table 4.10. One acceptable envelope is achieved at a threshold of 1500 (bearing in mind that thresholds need to be changed by large amounts in this program to overcome noise); the best area ratio then being the peak value in fig 4.23.

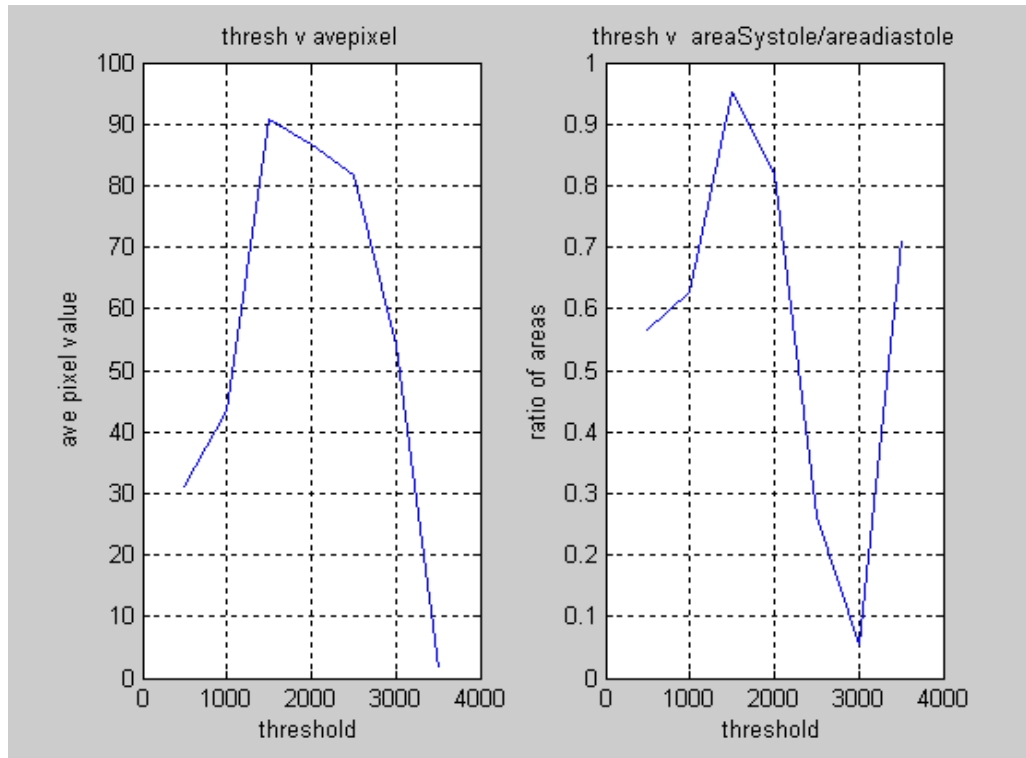


fig 4.23 Area ratio and average pixel value vs threshold for cycle one – differential method

threshold	500	1000	1500	2000	2500	3000	3500
R	0.566	0.626	0.951	0.819	0.262	0.053	0.711

table 4.10 Area ratio vs threshold for cycle one – differential method

The looped differential program can now be applied to the other three cycles on the image in fig 4.1. The graphs in fig 4.24 display area ratio and average pixel value versus threshold for the four cycles. It was noticed that for cycle one a threshold near 1500 produces an acceptable envelope curve. The same thing happens in the other three cycles. Thresholds for the differential method are spaced 500 apart due to noise in the original profile peaks. Table 4.11 shows the four values of area ratio at this threshold and the average (1.085). Thresholds of 1000 and 2000 show too much distortion in their envelopes. This program is rather sensitive to noise.

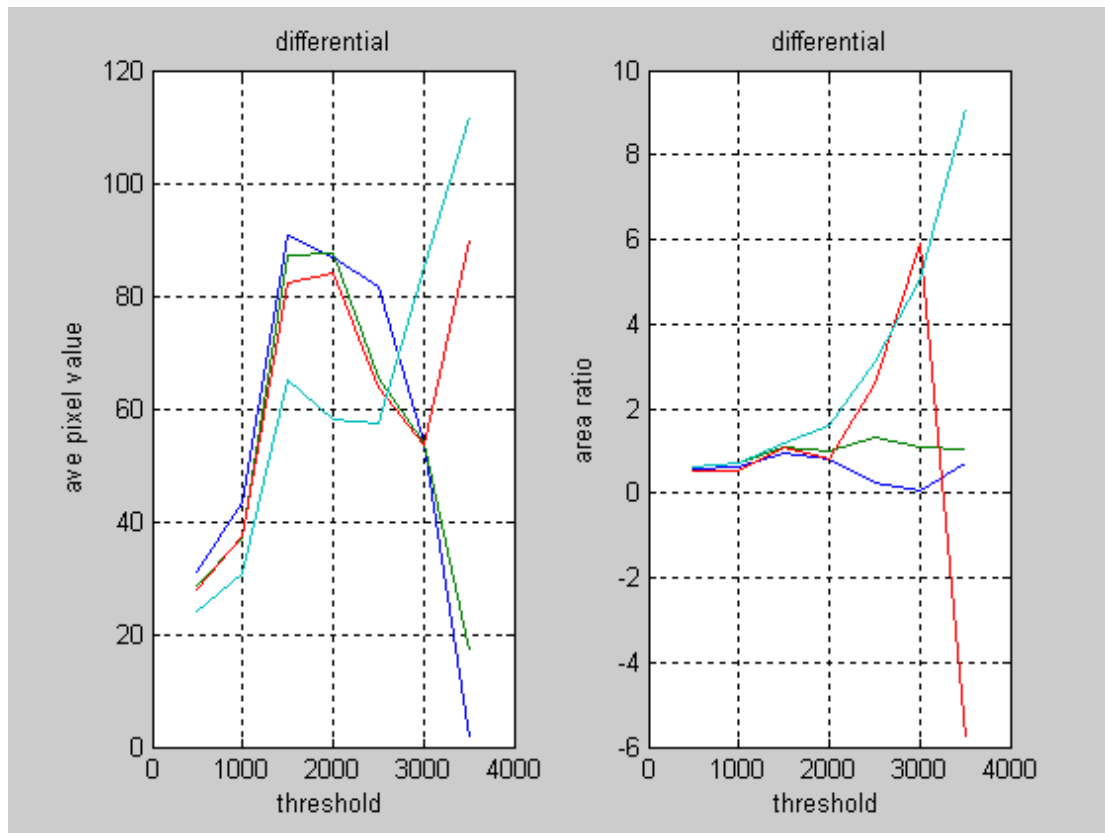


fig 4.24 Area ratio and average pixel value vs threshold for all four cycles in fig 4.1
Differential method

threshold	500	1000	1500	2000	2500	3000	3500
cyc1	0.566	0.626	0.950	0.819	0.261	0.053	0.710
cyc2	0.600	0.700	1.100	1.000	1.300	1.100	1.048
cyc3	0.525	0.548	1.091	0.806	2.576	5.903	-5.786
cyc4	0.600	0.700	1.200	1.600	3.100	5.050	9.110
Rave	0.573	0.644	1.085	1.056	1.809	3.027	1.271

table 4.11 Area ratios for all four cycles in fig 4.1 and average for each threshold

4.2.4 Percentile method

Three clicks are made on cycle1 in fig 4.1. The differential program is looped ten times to produce smoothed envelope curves for the thresholds 5%, 10%, 15%, 20%, 25%, 30%, 35% (see section 2.5). Fig 4.25 and 4.26 show the envelope curves. A threshold of 45% (55% of total pixel value lies under the curve) gives an acceptable envelope. At this threshold the area ratio is a little less than the values of the other programs. Table 4.12 gives the area ratio for each threshold and graphed in fig 4.27.

Researchers⁶ using this method tend to use thresholds of 5% to 10%. The reason for this is that the peak value of systole will be closer to the value in the original image. Note that at a thresholds of 40% to 50% although the envelope is acceptable and the area ratio comparable the peak value of systole (and other points) are reduced. This program would have to be used with caution with dim images.

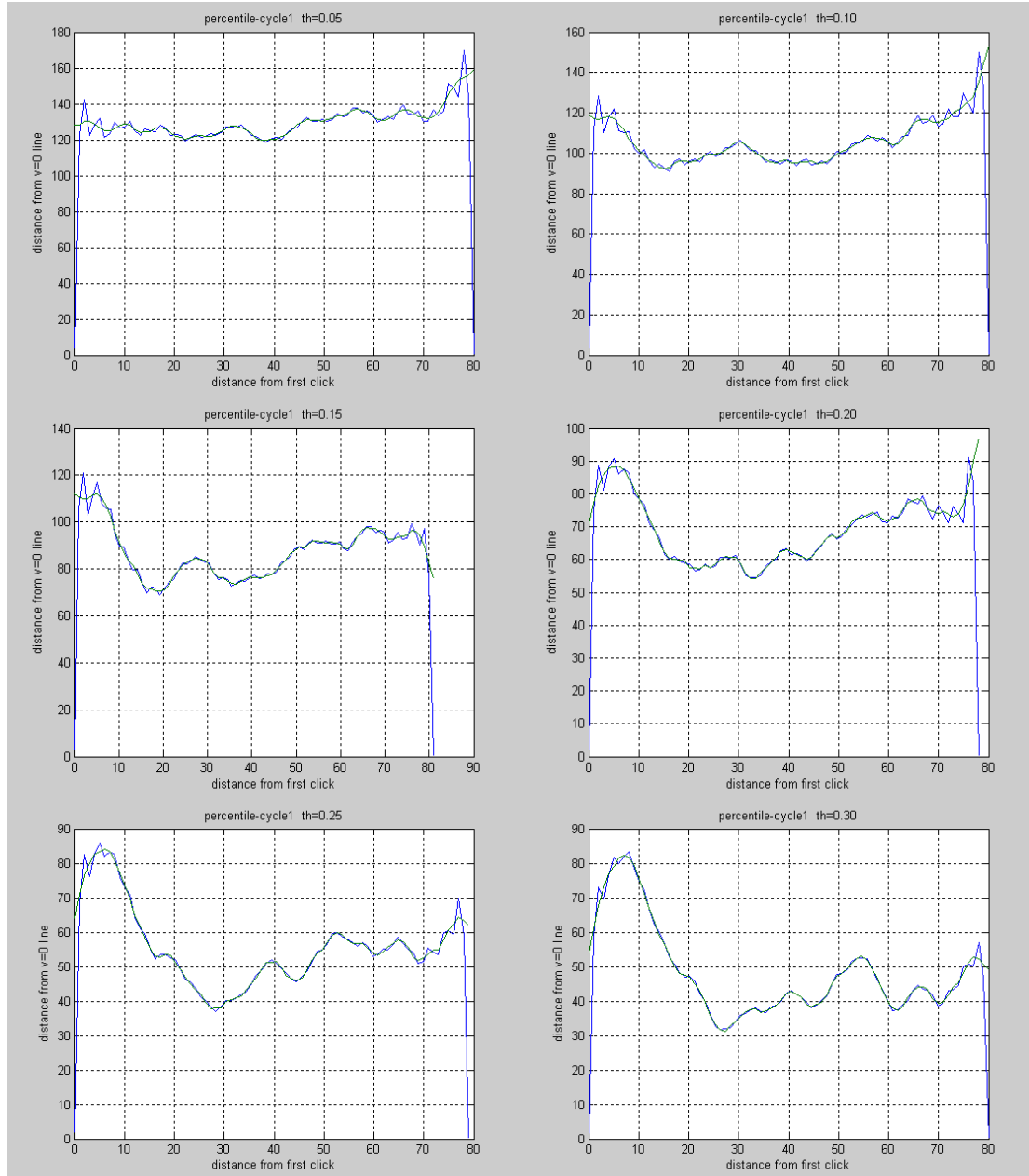


fig 4.25 Envelope curves for cycle one using the percentile method. Thresholds 5% - 30%

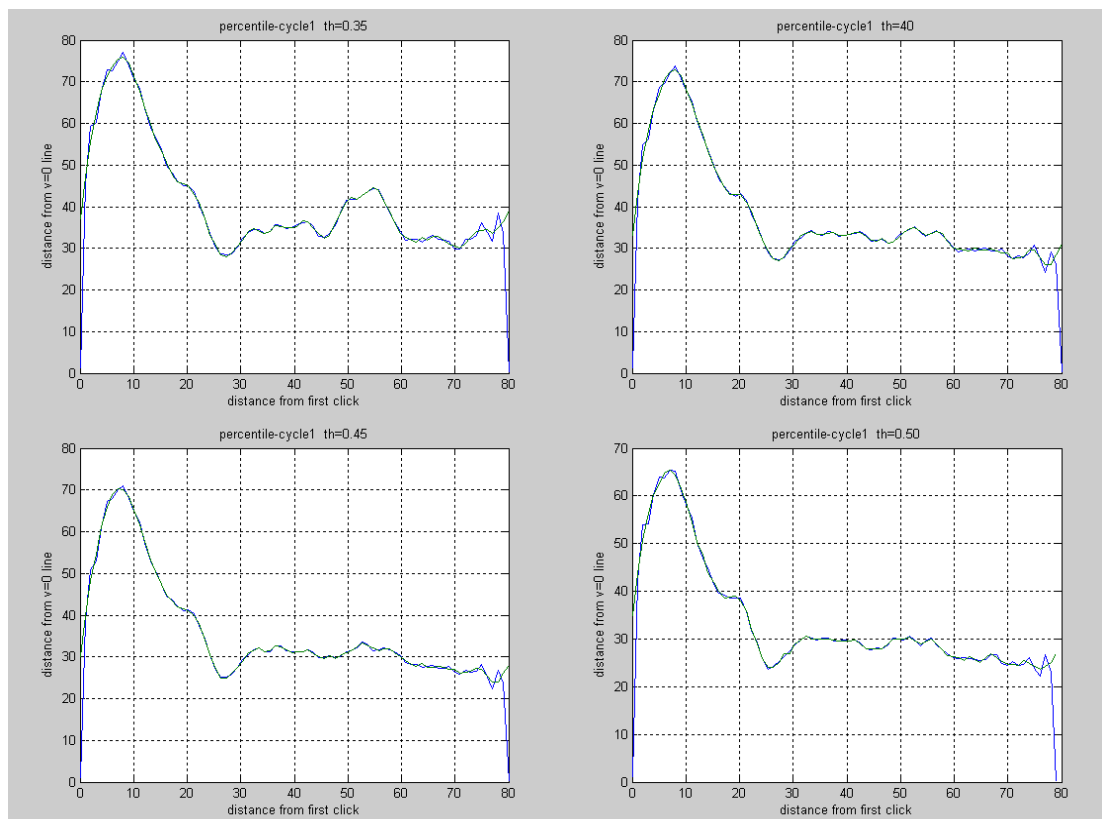


fig 4.26 Envelope curves for cycle one using the percentile method. Thresholds 35% - 50%

threshold	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
R	0.531	0.530	0.489	0.609	0.719	0.730	0.817	0.886	0.928	0.881
ave pix	37.57	48.01	60.66	73.43	91.61	112.5	134.3	143.7	148.8	134.8

table 4.12 Area ratio and average pixel value vs threshold using percentile method

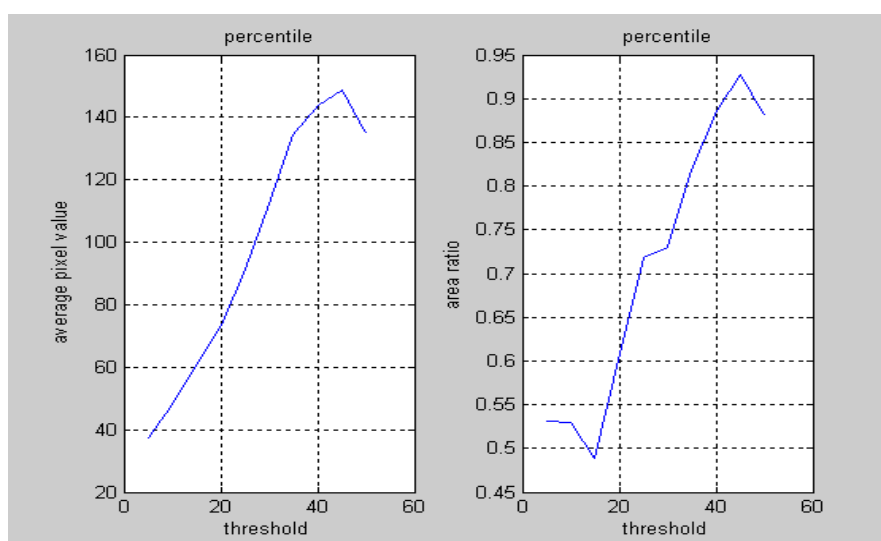


fig 4.27 Average pixel value and area ratio vs threshold for percentile method

The looped differential program can now be applied to the other three cycles on the image in fig 4.1. The graphs in fig 4.28, display area ratio and average pixel value versus threshold for the four cycles – the lowest line in the average pixel value graph is for cycle four.

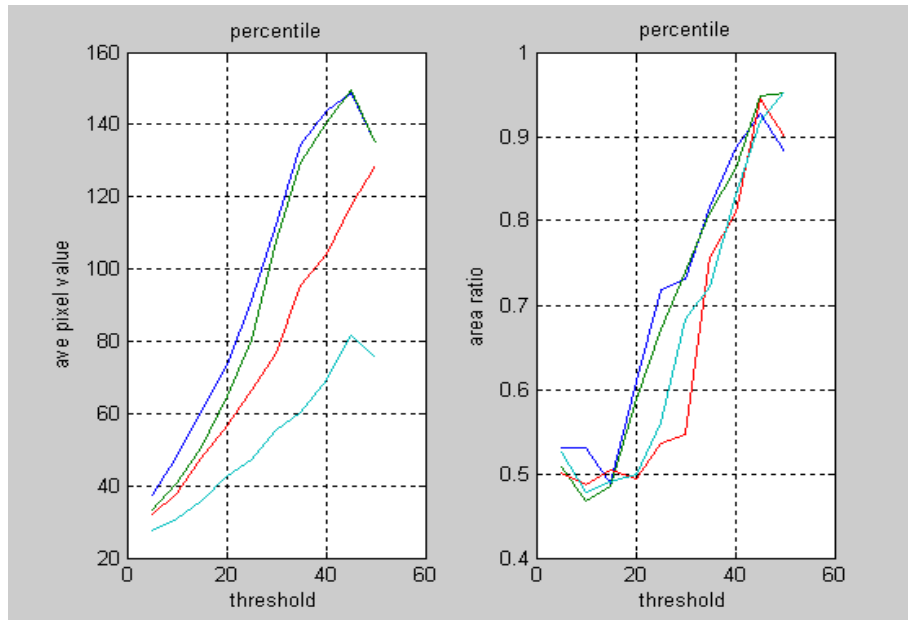


fig 4.28 Average pixel value and area ratio vs threshold for all four cycles in fig 4.1

threshold	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
R_{ave}	0.52	0.49	0.49	0.53	0.62	0.67	0.78	0.85	0.93	0.92

Table 4.13 Average area ratio for each threshold for the four cycles

4.2.5 Comparison summary

It is necessary for the syncope study to arrive at a mean area ratio for the cycles evaluated in each image. Choice of threshold is dependent on background noise in a person's set of images. Once this is chosen ideally it should be kept constant while evaluating all those images.

Table 4.14 summarizes area ratio and threshold for each of the four programs which will give an acceptable maximum velocity envelope when applied to the image in fig 4.1.

The ten-pixel average method gives the greatest flexibility in choosing thresholds. It is the quickest version to arrive at its area ratio result and preserves the height and shape of the original image. The edge method takes the longest time, giving area ratio results and waveform details similar to the ten-pixel-average method. *If over 1000 images are to be processed by this interactive method then the ten-pixel average version is the method of choice.* Note the percentile method processes images more quickly than the edge and differential method. This method has been widely used but as in the example just shown, noisy images require a high threshold - 40% or 50% of the pixel intensity between the maximum and minimum envelope curve. It is usual to work with 5% – 10%.

program	ten-pixel-average	edge	differential	percentile
threshold	70	50	1500	45%
area ratio	1.01	1.18	1.09	0.93

table 4.14 The four programs with ‘best’ threshold and average area ratio for the image

4.3 Comparing the four programs on a noisy dim image

Noisy images with their systolic/diastolic cycles very dim occur in almost all of the person-image sets. Fig 4.29 shows a typical image of this sort. The program chosen to evaluate all images in the study will have to be able to process these images and obtain a meaningful result. The cycle with A on its systolic peak and B at the end of the diastolic region is reasonably well formed with both regions having similar pixel values.

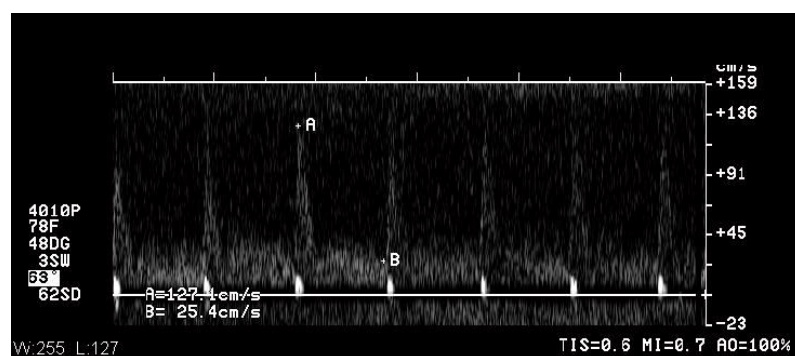


fig 4.29 Noisy image with dim pixel values

4.3.1 Ten-pixel-average method

Three clicks are made on the brighter cycle (third from the left) and the ten-pixel-average method looped ten times to produce maximum velocity envelopes for thresholds 30, 40, 50, 60, 70, 80, 90,100. The envelopes for thresholds 40 to 90 are shown in fig 4.30. In the original image (fig 4.29), the height of systole is 153 (pixel intensity 58) and the maximum of diastole 50.

A threshold of 50 (top right graph in fig 4.30) preserves the height of systole and diastole for this cycle with an area ratio just less than 0.9 (fig 4.31). There is a marked decrease in systolic height for thresholds 60 and 70; these thresholds producing reasonable envelopes. However their area ratios are significantly less.

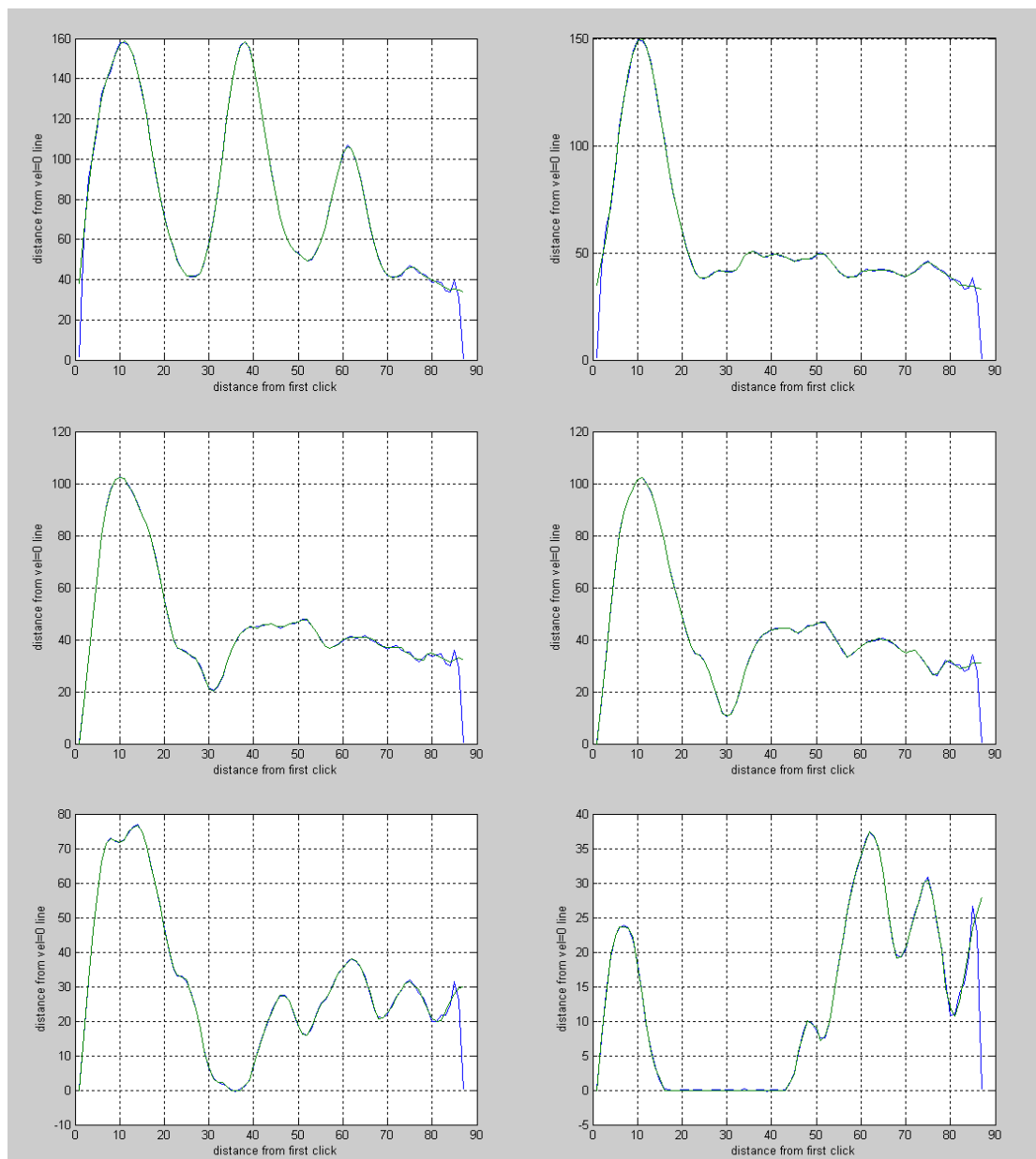


fig 4.30 Maximum velocity envelopes using the ten-pixel-average method. Thresholds 40 to 90 in steps of 10

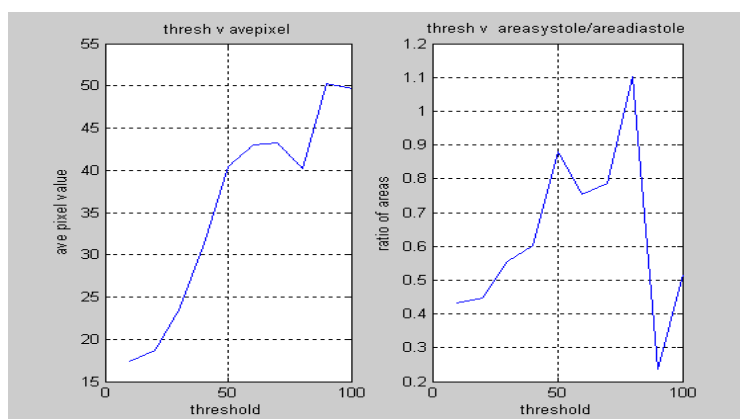


fig 4.31 Average pixel value and area ratio vs threshold for ten pixel-average method

4.3.2 *Edge detection method*

Three clicks are made on the same cycle as the last section in fig 4.29 and the edge method looped six times for thresholds of $m=30, 40, 50, 60, 70$ and 80 . Edge images for thresholds 30 and 40 are shown in fig 4.32 the others being rather noisy.

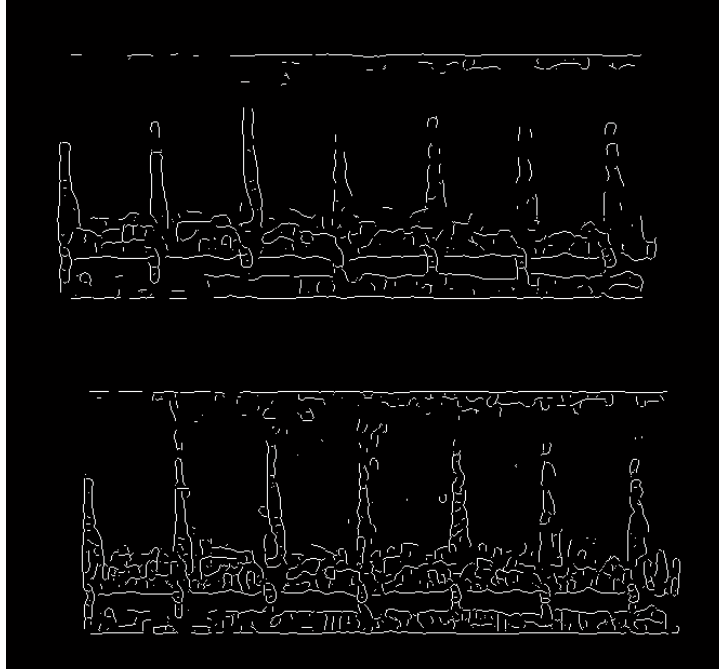


fig 4.32 Edge images for thresholds $m=30$ and $m=40$ using the edge detection method on the image in fig 4.29

The third cycle maximum velocity envelopes for these two thresholds are shown in fig 4.33. A threshold of $m=30$ produces a good systolic envelope with height over 160 (153 as noted from `pixval`). However its diastolic region is distorted due to noise in the image.

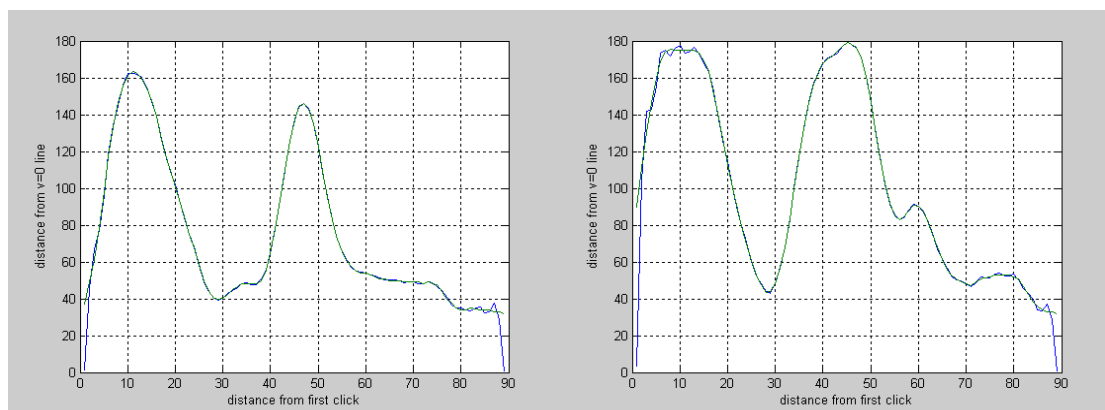


fig 4.33 Maximum velocity envelopes for thresholds $m=30$ and 40 for the images in fig 4.32

Because the edge image for threshold $m=30$ has noise over the diastolic region for the third cycle it may be that going to a threshold of $m=20$ may eliminate this. Fig 4.34 shows that while the noise is avoided the top edge of systole has also been removed leading to a drastic distortion of the envelope in the systolic region and reduction in height.



Fig 4.34 Edge image for $m=20$ and maximum envelope for this threshold

As a general comment; edge detection for $m=30$ and $m=40$ produces well defined outlines even though the original image is dim! Another method of reducing the noise after edge detection could make this edge method of generating a maximum velocity very useful if the time to produce it is not a factor⁷.

4.3.3 Differential method

Again three clicks are made on the third cycle in fig 4.29 with letters A and B written on it and the differential method looped seven times for thresholds 500, 1000, 1500, 2000, 2500, 3000 and 3500. The maximum velocity envelopes for thresholds 1000, 1500, 2000 and 2500 are shown in fig 4.35. This method does not generate acceptable envelopes for this image with low intensity pixel values.

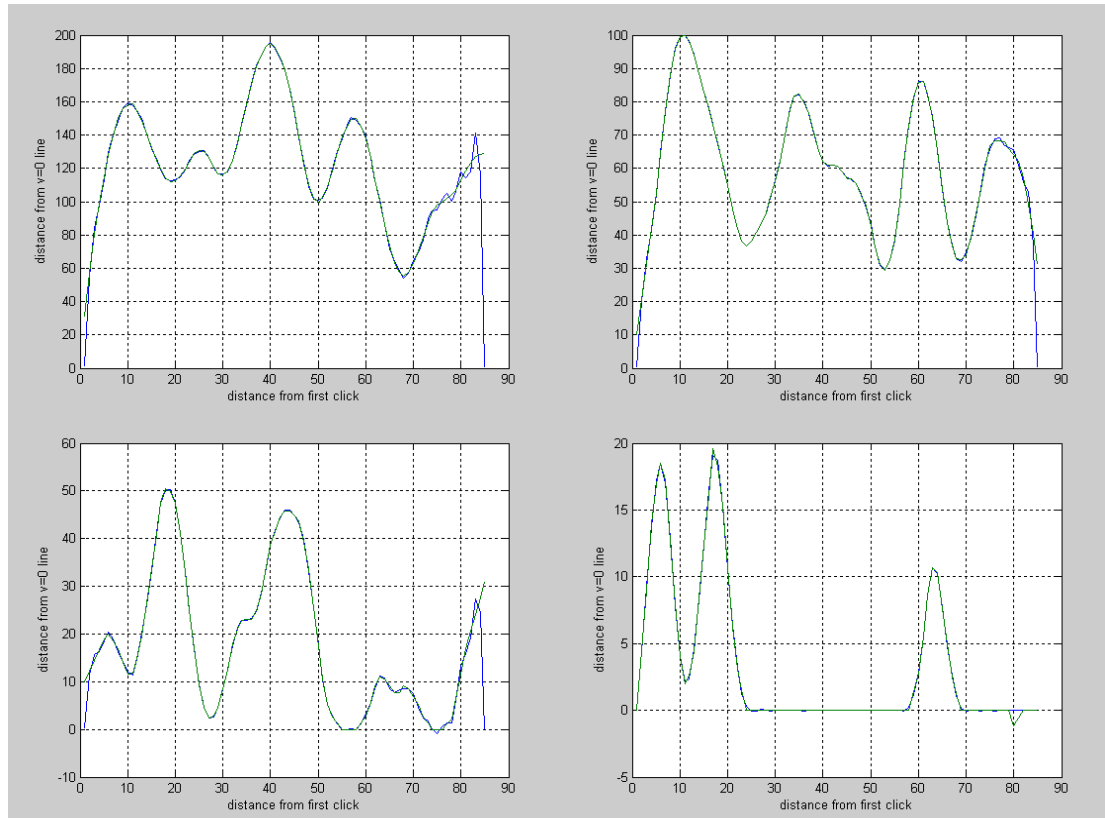


fig 4.35 Maximum velocity envelopes for thresholds 1000 to 2500 for the differential method

4.3.4 *Percentile method*

The percentile method is looped ten times after three clicks are made on the third cycle in fig 4.29. The seven thresholds are; 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45% and 50%. Fig 4.36 shows the envelopes for thresholds 5% and 10%. The percentile method does not produce an acceptable maximum velocity envelope for this dim image at any of the ten thresholds.

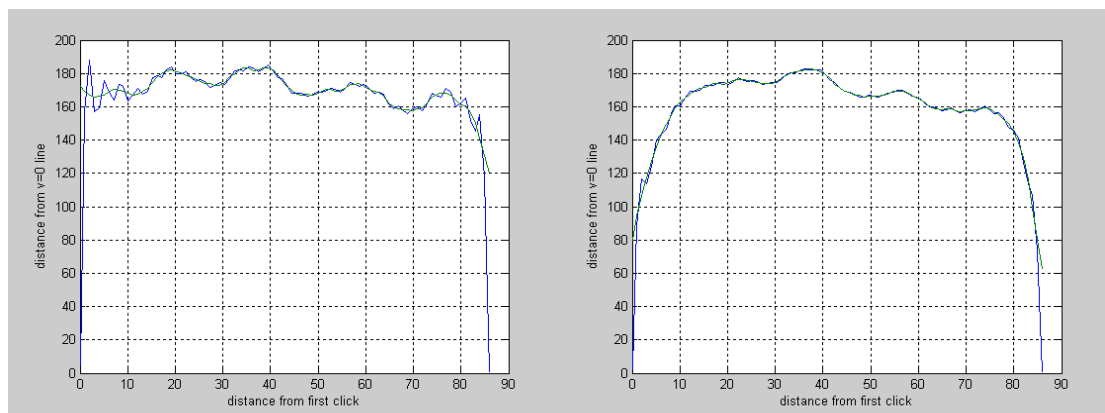


fig 4.36 Maximum velocity envelopes for thresholds 5% and 10% for the percentile method

4.3.5 Comparison summary

Since an image like the one in fig 4.29 occurs in most image sets it is important to be able to extract a maximum velocity envelope for two or more of the cycles in the image. The ten-pixel-average method is able to generate quickly a maximum velocity envelope for the cycle in this image at a threshold of 50 which is the threshold used for all of the images in the set that this belongs to (and consistently for other image sets). The edge method produces well-formed edges with noise also present at the threshold used for the rest of the images in the set. More filtering code could be incorporated into the program to deal with this ⁸. However the edge method takes longer to generate a maximum velocity than the ten-pixel-average method and including more code to remove noise after the edge image is formed will increase the time. In general the differential and the percentile methods do not produce acceptable maximum velocity envelopes for noisy dim images. If the percentile method does manage to produce one its threshold is in the region of 45%; meaning it has penetrated deeply into the region of interest reducing the height of systole and diastole.

4.4 A complete image set compared

To complete this chapter on comparing the four programs, a complete image set is analysed using just one cycle from each image. Person six in the study has just sixteen images and the cycle chosen already has the marks A on systole and B at the end of diastole and best represents the systolic/diastolic cycles for the image. The complete syncope study will have two or more cycles used for each image and the area ratio averaged. Choosing only one cycle will give an *indication* of how the four versions compare over a complete image set but not take a long time as if averaging more than one cycle.

The thresholds for each method are as in the table 4.14: Ten-pixel-average (70), edge (50), differential (1500) and percentile 45%. The area ratios generated by the four programs for the sixteen images are shown in table 4.15 and graphed in figs 4.37 and 4.38. The area ratios are plotted against image number indicating the change in the ratio as images are captured during supine and tilt positions. Fig 4.37 compares the ten-pixel-average and edge methods. For every image number the edge method generates a higher area ratio - the average difference is 0.37 for the thresholds used. Both versions tend to follow the same pattern of change with noticeable change near the beginning and after the tenth image. Fig 4.38 compares the differential and percentile methods. Both seem to follow the same pattern of change as the other two versions but noisy images at 4 and 14 have produced distorted envelopes and hence area ratios that are significantly different.

The change in area ratio with image number is of interest for the fainting study rather than the absolute value for each image. The sixth column in table 4.15 lists the Loupas ratio (outlined in section 3.3) for each image, using the ten-pixel-average method. Fig 4.39 shows that it changes in a similar way to the other ratios, with greater range.

image	10-pix-ave	edge	differential	percentile	Loupas
1	1.5076	1.8656	1.6458	1.8025	5.3335
2	0.9061	1.3599	1.0333	1.0775	3.4046
3	1.0306	1.4748	1.0963	1.0984	4.0111
4	0.9331	1.3500	1.1000	0.5024	3.5481
5	1.0879	1.3152	1.2055	0.8395	3.7572
6	1.0849	1.1729	0.9748	1.1674	3.6154
7	1.0464	1.1942	1.0475	1.0210	3.6665
8	1.1187	1.5059	1.0698	1.1965	3.8042
9	1.0886	1.2813	1.1376	1.0328	3.7704
10	0.9619	1.2788	0.9064	0.9586	3.4988
11	1.2069	1.5009	0.9339	1.2386	5.3902
12	2.3621	2.9778	2.1415	1.6895	12.9581
13	2.0633	2.3604	2.2988	2.1403	8.4257
14	1.7477	2.4174	0	1.4957	10.4645
15	1.9317	2.2094	1.6588	2.0237	6.9732
16	1.4992	1.8407	1.3520	1.6795	5.6683

table 4.15 Comparison of area ratios for person 6 using the four methods

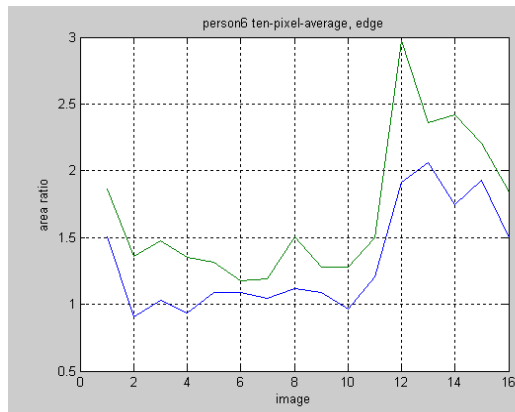


fig 4.37 Area ratio vs image number for ten-pixel-average and edge methods

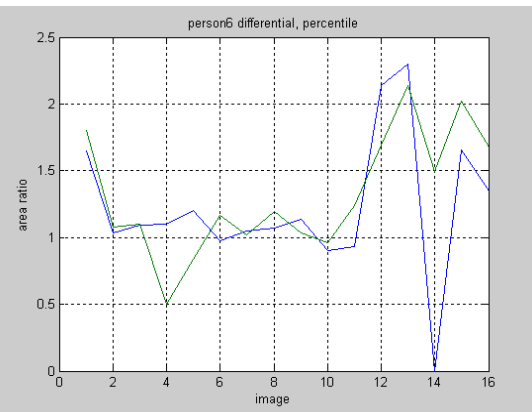


fig 4.38 Area ratio vs image number for differential and percentile methods

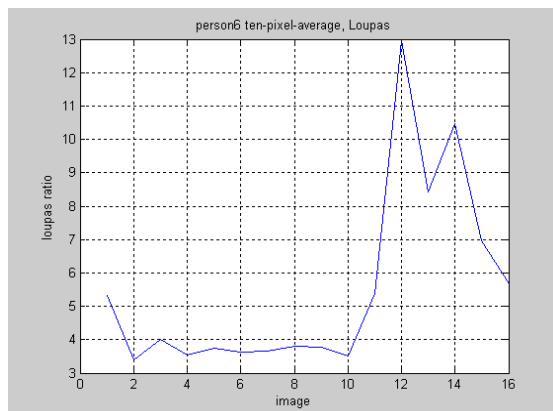


fig 4.39 Loupas ratio vs image number using ten-pixel-average method

CHAPTER FIVE

Summary and Conclusion

5.1 Maximum velocity envelope

As noted in the abstract for this thesis, the challenge for this project is to extract a maximum velocity envelope for over 1000 Doppler images in the syncope study. Since all image sets have some noisy images and some with one or more cycles missing or very dim, the Matlab program chosen out of the four tested, is the ten-pixel-average method. This program rapidly arrives at a smoothed maximum velocity envelope after points are selected from the image interactively. Thresholds for pixel averaging are chosen after viewing the whole image set to assess noise levels and the threshold level kept constant for all images in the set. After smoothing the envelope preserves the essential height and form of the cycle. Interactively choosing cycles in an image enables individual cycles to be selected thus avoiding poor quality systolic/diastolic cycles.

The edge detection method is a very good alternative to the ten-pixel-average method, taking more time to process individual cycles and possibly, more code will have to be added to eliminate noise captured in the edge image to maintain a constant threshold for the Gaussian filter.

5.2 Area ratios and indexes

After the maximum velocity envelope has been achieved, Matlab then fits a close fitting curve to the smoothed envelope. The least squares method used employs a Fourier series (equation (4)) taken to fifty components. The tight fitting curve adequately enables the area of systole, area of reverse flow and area of positive diastole to be calculated. More components in the Fourier series means more high frequency components but a quick check on area ratio versus component number shows that this reaches a constant value from 35 components onward (see fig 5.1).

Storing the three separate areas also means that when this data is related to the other measurements in the syncope study, different forms of the ratio can be investigated:

- a) Subtracting reverse flow from positive diastole – equation (6)
- b) Adding reverse flow to positive diastole – equation (7)
- c) Or the Loupas index; using Fourier coefficients and the average velocity in diastole – equation (8)

It is of course realized that a full Fourier transform would have arrived at the sum of component amplitudes however the best fit data has already been arrived at which is a very good approximation without adding any more code.

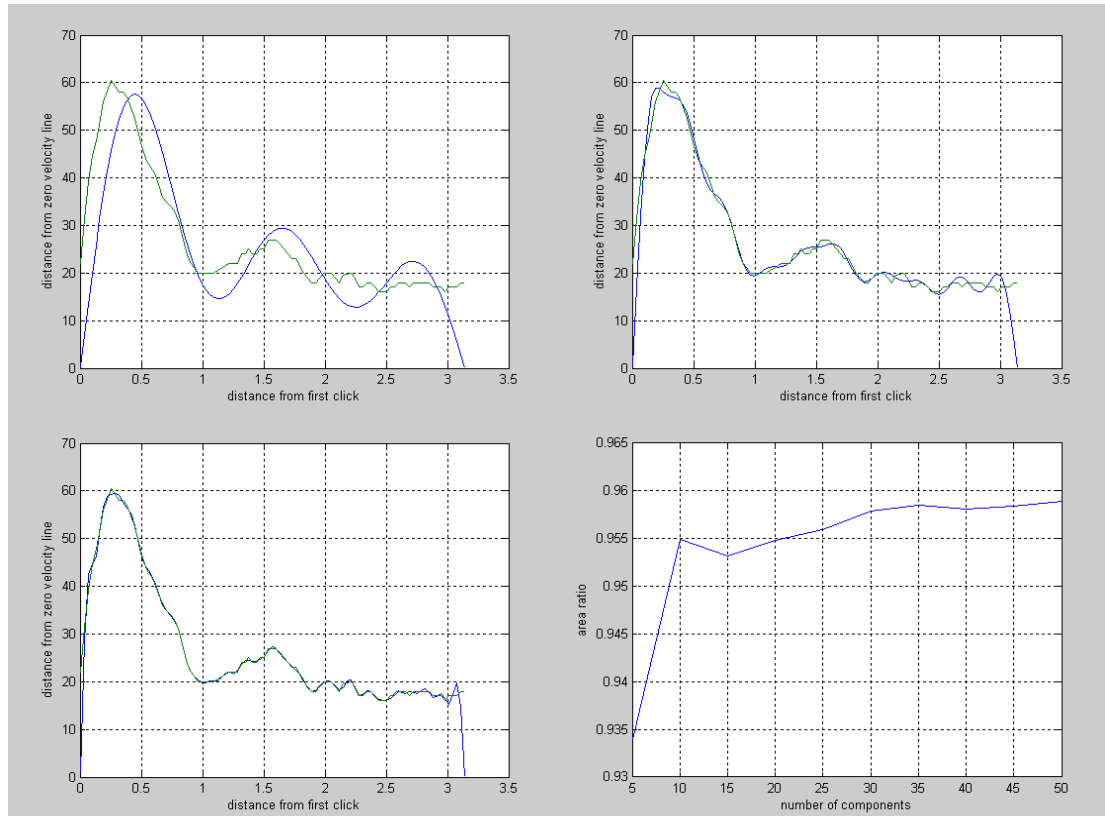


fig 5.1 Graphs showing curve fits for 5, 20 and 50 Fourier components and area ratio vs number of components.

5.3 Area ratio, heart rate, blood pressure and nerve activity

It is not the intention here to present the correlation (statistics) of an area ratio or index with the other measurements in the syncope study. Heart rate (beats/min), Blood pressure (mm-Hg) and muscle sympathetic nerve activity (MSNA bursts/min) are not available for all subjects at present.

The interplay of MSNA and the baroreceptor reflex in patients with vasovagal syncope is the subject of much research¹⁰. Studies of MSA baroreceptor sensitivity, heart rate and blood pressure under resting conditions and during a postural stress test, like the one in reference 10, have shown that MSNA is significantly decreased during the stress tests (syncope). The important inclusion in the syncope study at Christchurch hospital (hopefully to be printed in a medical journal) is the relationship between these measurements and a better evaluation of Doppler-wave-form changes (particularly diastole) in patients with vasovagal syncope using the quick and effective ten-pixel-average method.

The ten-pixel average method has been used to evaluate all images in the syncope study and the three areas mentioned in section 5.2 stored. As an *indication* of the relationship between the area ratio (equation (6)) and heart rate, blood pressure and MSNA measurements to hand, person 15 and person 22 results are presented below.

Table 5.1 shows the area ratio and other measurements listed with image number for person 15. This subject was tilted to 30° just before image 7 and then to 60° just before image 18.

image	arearatio	HR	BP	MSNA
1	0.984495		61	103
2	0.876963		56	103
3	1.056775		58	103
4	1.032347		56	101
5	0.934923		58	103
6	1.088385		62	104
7	1.082348		58	103
8	1.508478		67	110
9	1.366278		71	113
10	1.459642		70	113
11	1.011489		71	111
12	1.009462		69	110
13	0.968983		71	110
14	0.865092		75	110
15	0.874514		72	111
16	1.061583		67	108
17	0.877961		66	105
18	0.923263		78	104
19	0.911725		87	106
20	0.952074		82	105
21	0.834452		87	108
22	1.048274		85	106
23	1.072131		85	106
24	0.927749		86	102
25	0.918408		84	102
26	1.182037		90	100
27	1.188678		100	104
28	1.365276		102	102
29	1.365276		87	61
30	6.786974		66	72
31	6.469941		56	88
32	2.458645		54	92
33	1.465054		59	98
34	1.27695		61	98
35	1.053467		57	100
36	1.137238		61	101
37	1.000439		59	102

table 5.1 Area ratio, heart rate, blood pressure and nerve activity for person 15

Fig 5.2 shows the graph of area ratio plotted against image number and fig 5.3 shows area ratio, heart rate, average blood pressure and MSNA plotted against image number.

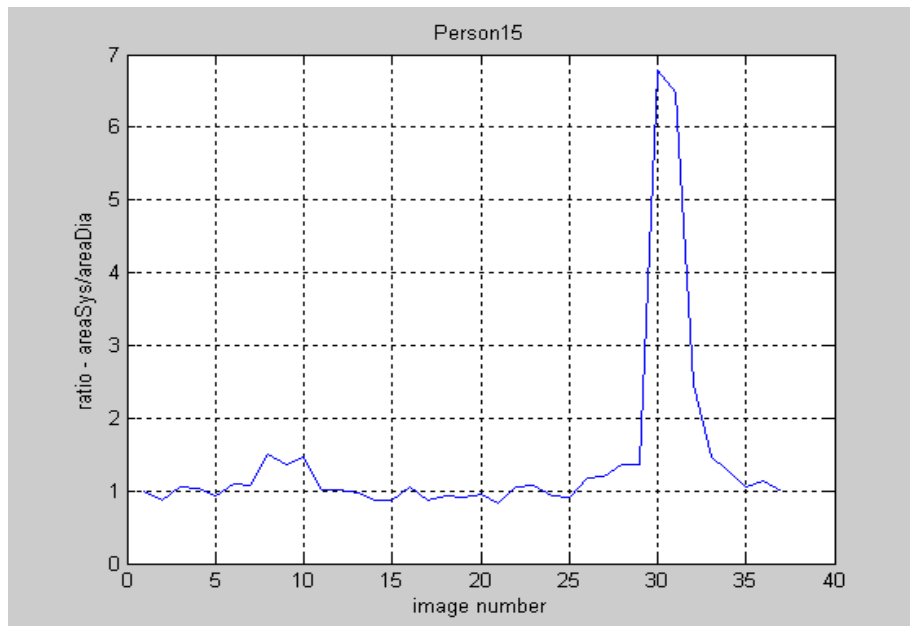


fig 5.2 Area ratio plotted against image number for person 15

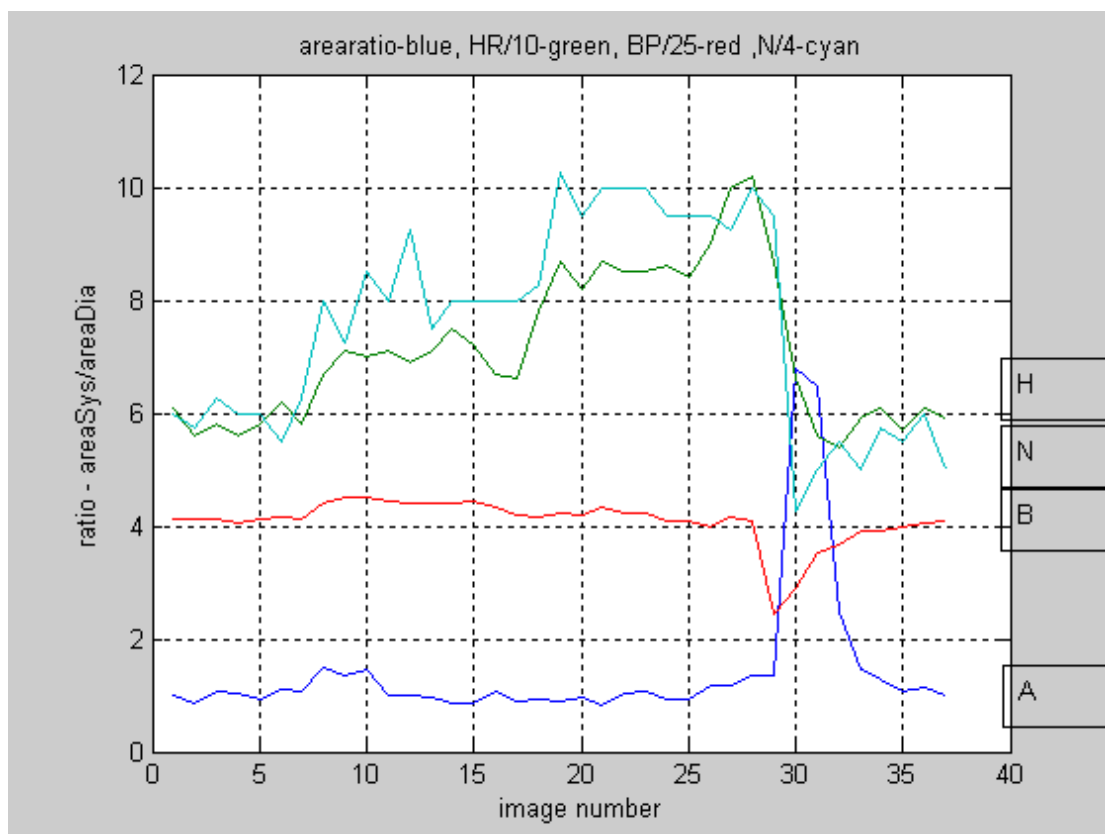


fig 5.3 Area ratio, heart rate, blood pressure and nerve activity plotted against image number

Beside each curve is an identification letter:

- A, area ratio
- B, blood pressure
- H, heart rate
- N, nerve activity

To bring the measurements to the same scale as the area ratio, blood pressure readings have been divided by 25, heart rate measurements by 10, and nerve activity by 4. Bearing in mind this scaling, heart rate, blood pressure and nerve activity show a tendency to increase after the subject has been tilted at 30° and then to 60° with small changes in area ratio; then the three measurements sharply decrease near image 30 when the subject faints.

Fig 5.4 shows a Doppler image at the beginning before tilt and the Doppler image at position 30.

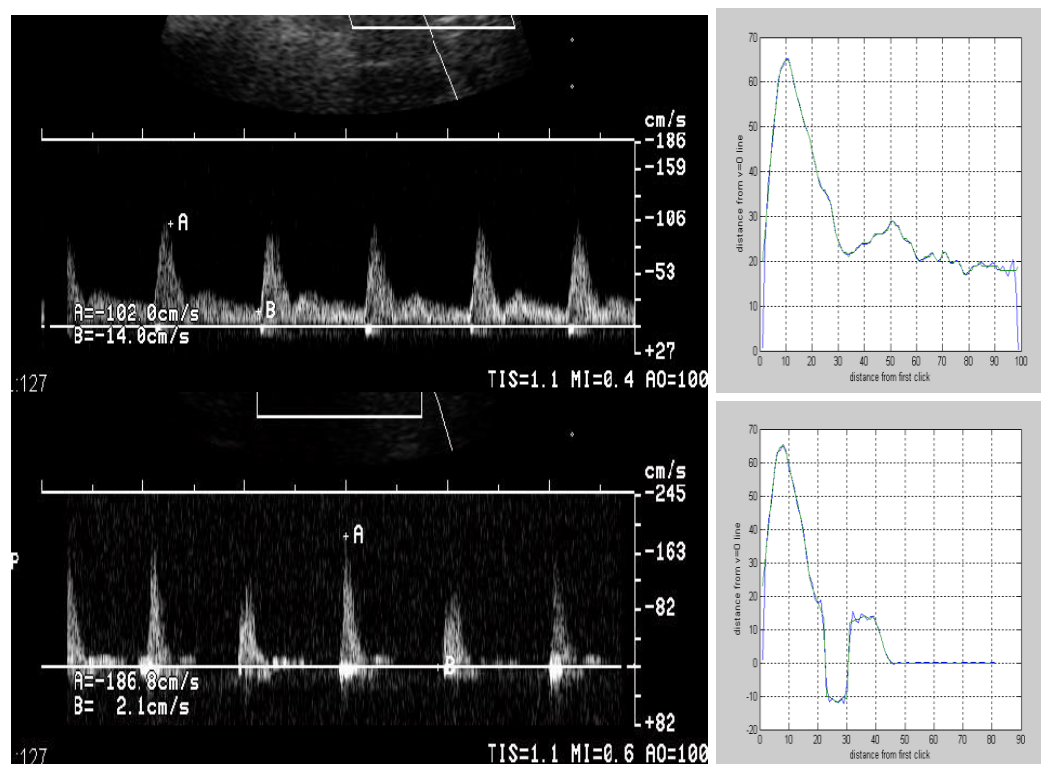


fig 5.4 Doppler images and waveforms for person 15 before tilt and at syncope

Reverse flow does not exceed positive diastole at image 30 (lower image in fig 5.4). Because it is subtract from positive diastole the total diastolic area is small so the area ratio is large. There is significant decrease in diastolic flow during syncope which indicates increased resistance to blood flow into the mesenteric artery; hence blood may be pooling in some other part of the lower body.

Person 22 in the study is one of the 'normal' subjects. Fig 5.5 shows the graph of area ratio and the other syncope measurements plotted against image number. Table 5.2 lists the syncope measurements with area ratio and image number. While changes have taken place in heart rate, blood pressure and MSNA after tilt (image 7), the area ratio does not show significant change.

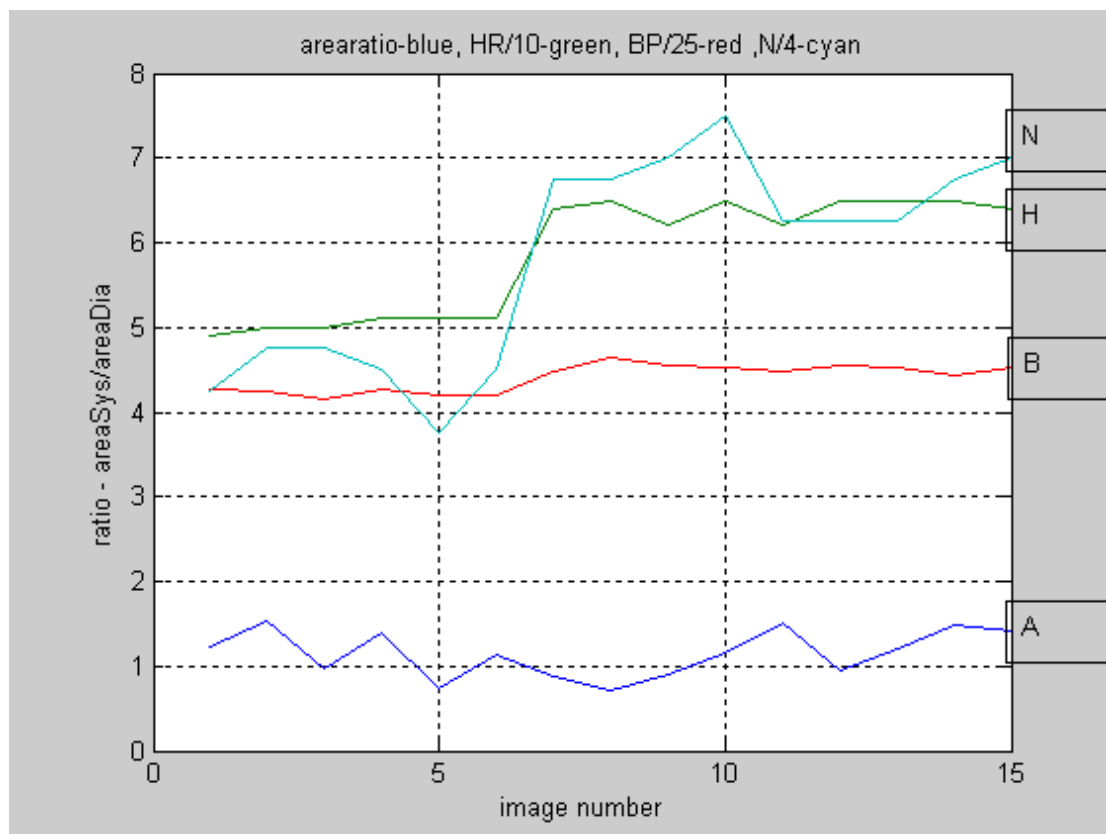


fig 5.5 Area ratio (A), blood pressure (B), heart rate (H) and nerve activity (N) for person 22

image	arearatio	HR	BP	MSNA
1	1.227489	49	107	17
2	1.542189	50	106	19
3	0.973464	50	104	19
4	1.397503	51	107	18
5	0.735406	51	105	15
6	1.145714	51	105	18
7	0.874375	64	112	27
8	0.710282	65	116	27
9	0.895552	62	114	28
10	1.151534	65	113	30
11	1.513276	62	112	25
12	0.94782	65	114	25
13	1.196833	65	113	25
14	1.483767	65	111	27
15	1.415898	64	113	28

table 5.2 Area ratio, heart rate, blood pressure and nerve activity data for person 22

Although the other syncope measurements are not available for person 1 in the study it is interesting to note the change in area ratio versus image number for this subject.

Fig 5.7 shows the graph for this person. At image number 13 the area ratio becomes negative. This is the faint point for person 1. Fig 5.6 shows the Doppler image at number 13 and the maximum velocity envelope. Positive diastole has almost disappeared and because reverse flow is subtracted the area ratio becomes negative. Again the significant reduction in diastolic flow at fainting tends to suggest an increase in arterial resistance at this point.

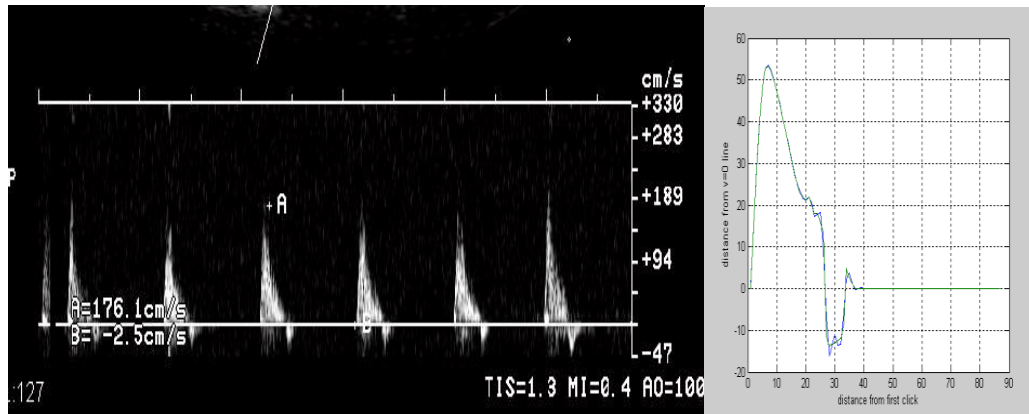


fig 5.6 Doppler image and maximum velocity envelope at point 13 (faint) for person

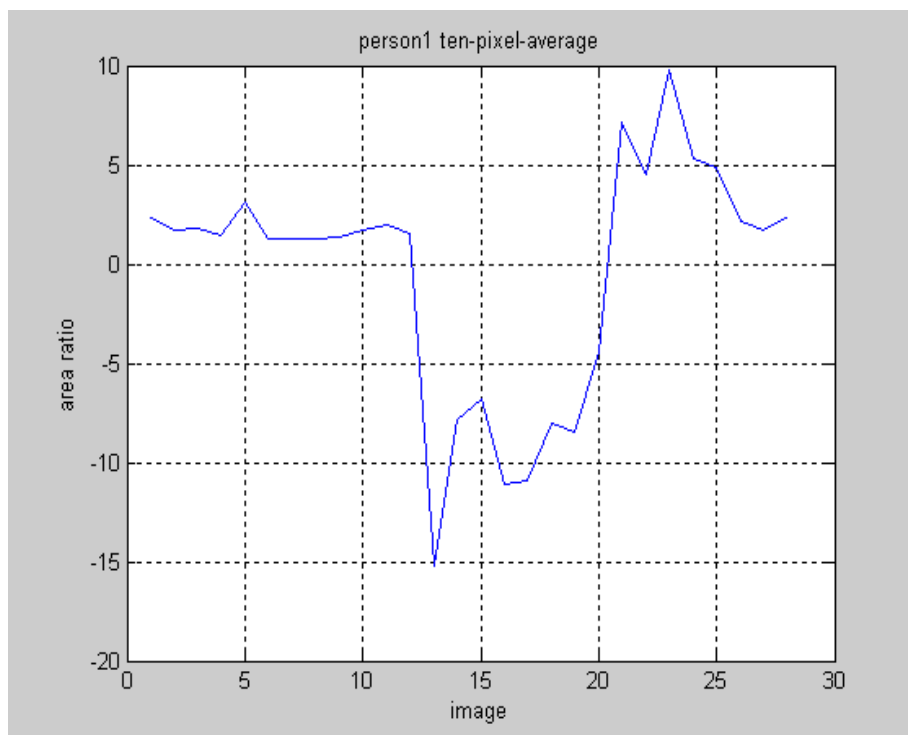


fig 5.7 Area ratio versus image number for person 1

5.4 Conclusion

In conclusion; ultrasound images of blood flow in the mesenteric arteries can be reliably analysed using a ten-pixel average method. This analysis permits images with varying amounts of noise to be compared using an area ratio obtained from a maximum velocity envelope. This area ratio is able to easily discriminate blood flow changes by taking into account early as well as late diastole in a sequence of images. The goal of this project has thus been met and the area ratio sequence for forty image sets available to be statistically correlated to other measurements made in the syncope study at Christchurch Hospital.

REFERENCES

- 1 *Artificial Neural Network Analysis of Doppler Shift Signals*,
Isabel A Wright, PhD 1998 .

- 1 *Modelling cerebral blood flow control during posture change from sitting to standing*. Olefsen M, Tran H, Ottensen J.
Cardiovascular Engineering: Vol. 4 No 1, March 2004

- 2 *Online identifying and quantifying Doppler ultrasound waveforms*.
Johnstone KW, Kassam M, Cobbold RS.
Med. and Biol. Eng. & Comput. 1982. 20, 336-342.

- 4 *Determination of the envelope function (maximum velocity envelope) in Doppler ultrasound flow velocity diagrams*
Tschirren J, Lauer RM, Sonka M
The University of Iowa, Department of Electrical and Computer Engineering. www.spie.org/scripts.

- 5 *Levenberg-Marquardt Algorithm*, A Miller
www.mathcom.com/nafaq/q230.2.1.html

- 6 *Doppler ultrasound physics, instrumentation and signal processing*,
Evans DH, McDicken WN. Wiley 2000.

- 7 *Doppler sonography in vascular diagnosis*.
ELCAT Medizinische Systeme. www.elcat.com

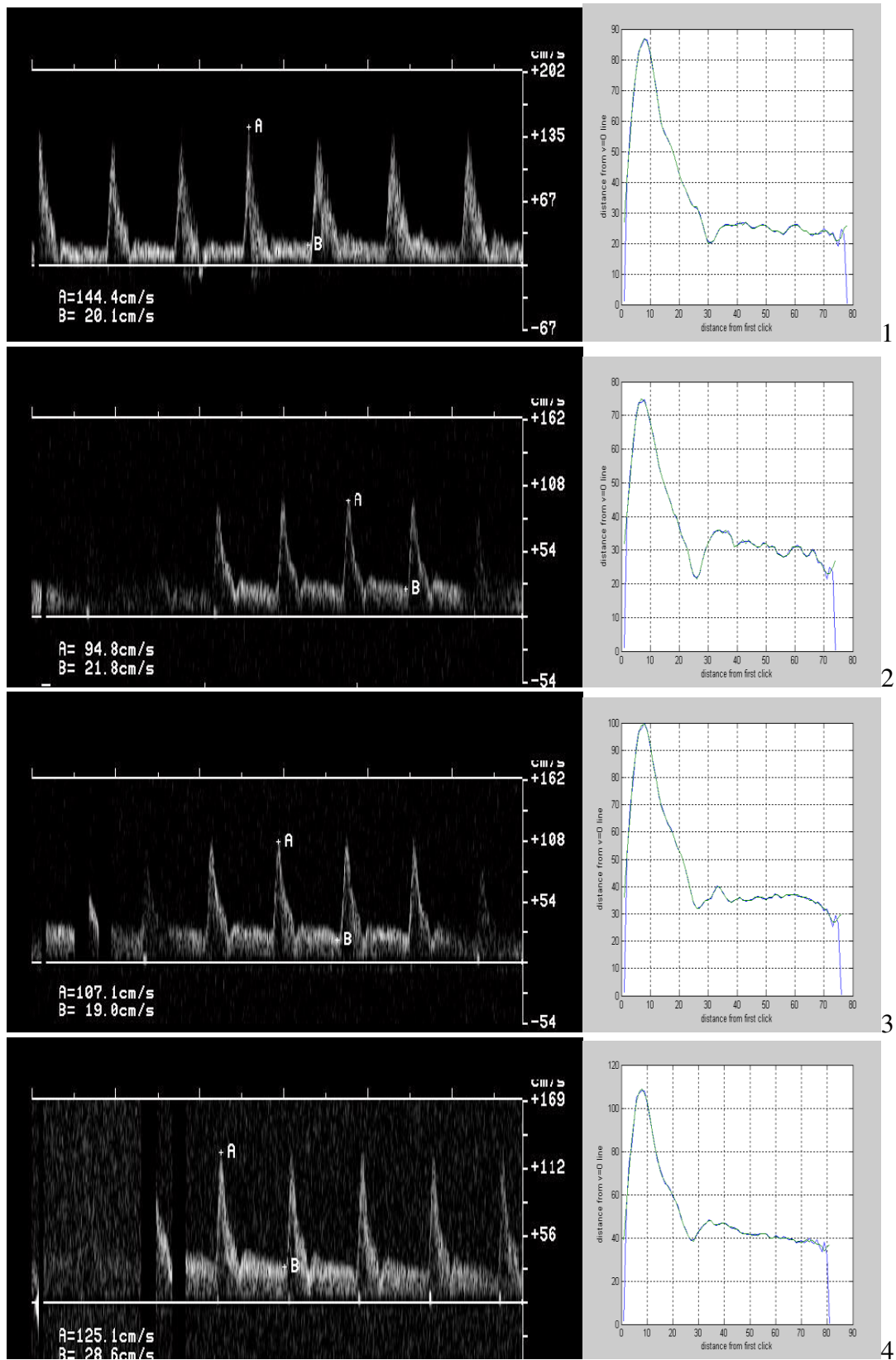
- 8 *Computer analysis and pattern recognition of Doppler blood flow spectra for disease classification in the lower limb arteries*.
Biomedical Engineering Laboratory, Clinical Research Institute of Montreal.
Allard L, Langlois YE, Durand L, Roederer GO, Beaudoin M
Cloutier G, Roy P, Robillard P.

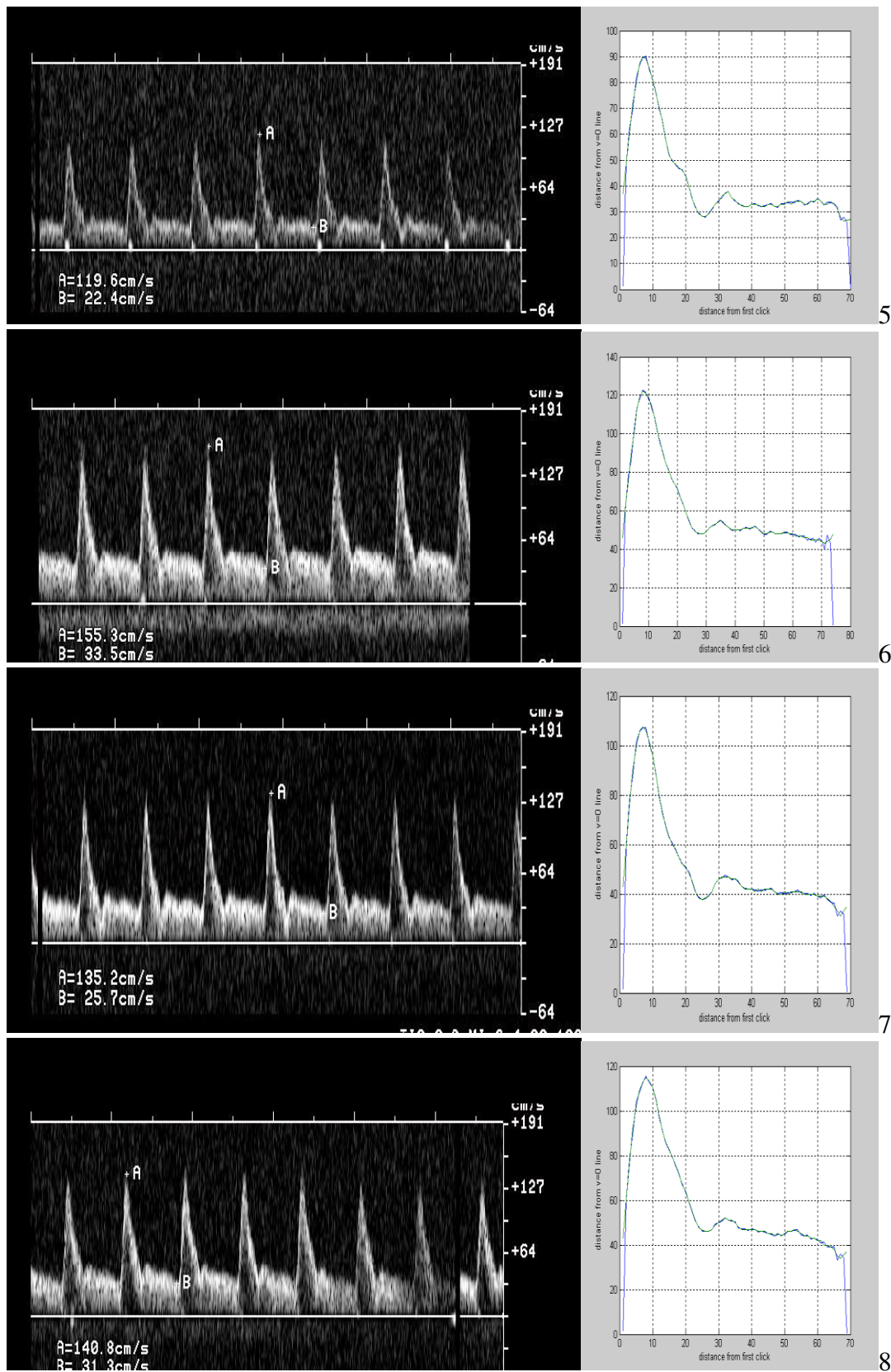
- 9 *PHYSBE, A physiological simulation of benchmark experiment using matlab simulink*. [www Mathworks.com](http://www.Mathworks.com)
First published by J McLeod in *Simulation* Vol.7 No.6 324-329 1966.

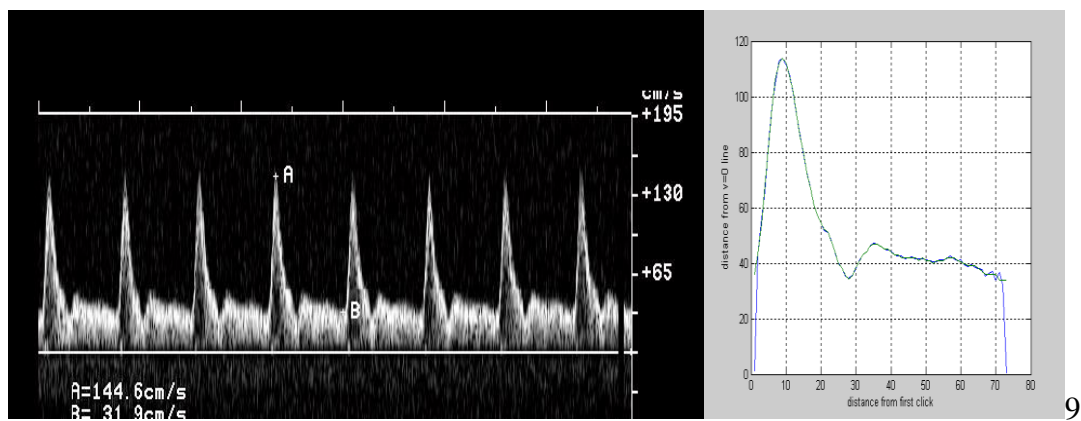
- 10 *Dysfunctional baroreflex regulation of sympathetic nerve activity in patients with vasovagal syncope*.
Bechir M, Binggeli C, Corti R, Chenevard R, Spieker L, Ruschitzka F, Luscher T, Noll G. *American Heart Association, Circulation* 2003; 107-1620. www.circ.ahajournals.org

APPENDIX 1

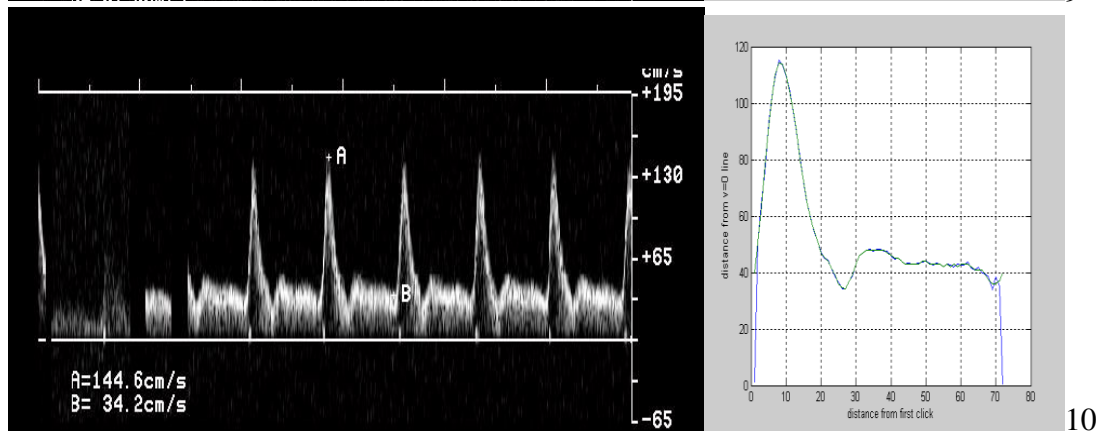
Doppler images (16) and maximum velocity envelopes for person 6.
Threshold set at 70 due to noise level in image 5 and 6 (kept constant).



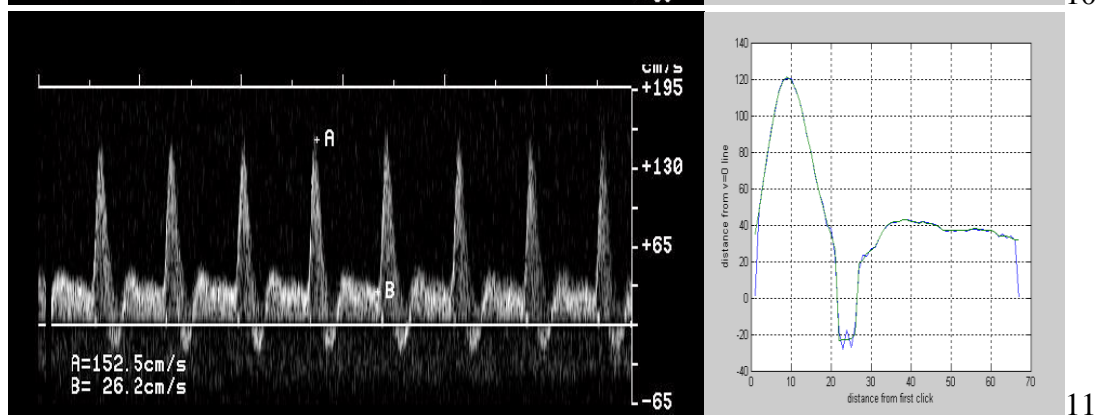




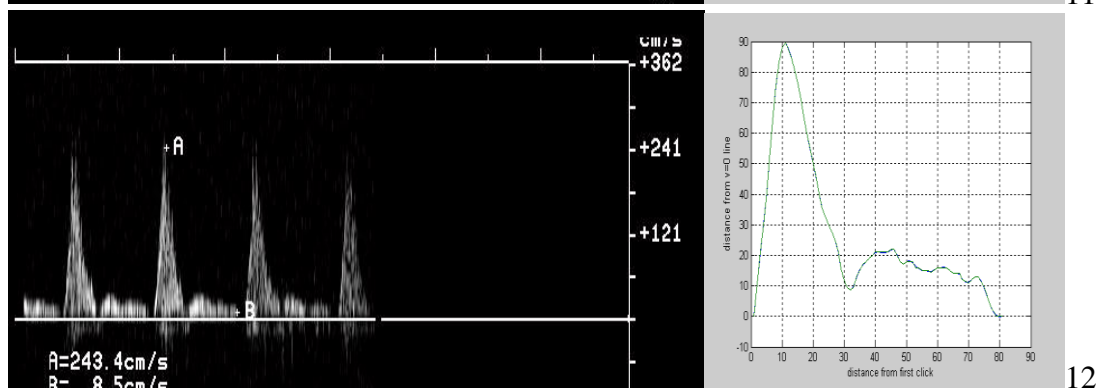
9



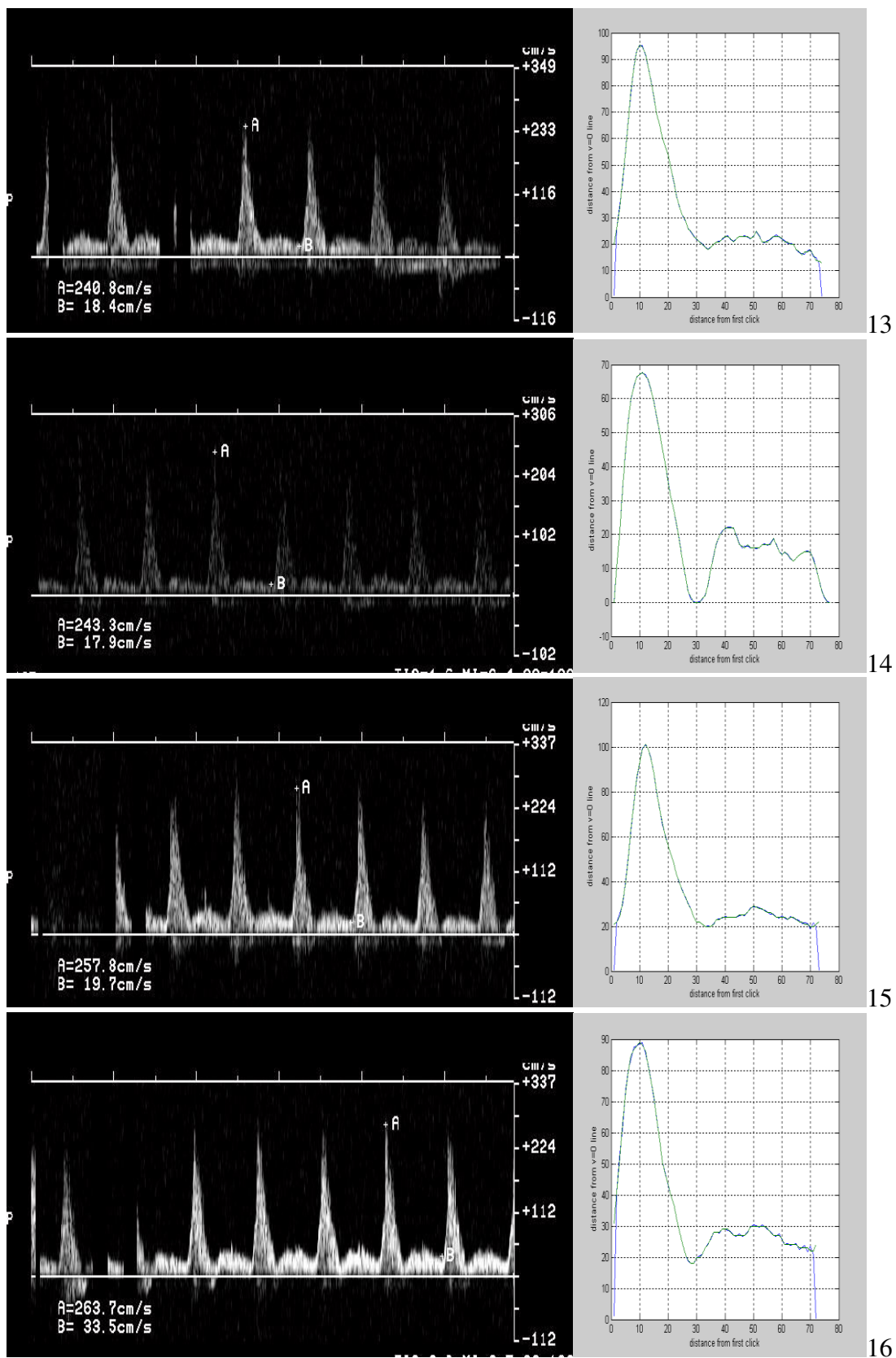
10



11



12



APPENDIX 2

1. Matlab script for ten-pixel-average method

```
% ten-pixel-average
clear all
%-----
%locate and display image
file=uigetfile('*.tif');
A=imread(file);
Ag=rgb2gray(A);
Ag(1:278,1:768)=0;
figure;imshow(Ag,'notruesize');[ca r p]=impixel;%ca code for clicks
%-----
%scan image to find j value of top line from first 5 scans
n=5;
for k=1:n
x=[325+k 325+k];y=[290 535];P=improfile(Ag,x,y);
for j=1:length(P);B(k,j)=P(j)';end
%u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
            C0=j;break;%break so get top line
        end;end;end
%-----
%scan image to find j value of V=0 line from first 5 scans
n=5;
for i=1:n
x=[325+i 325+i];y=[290 535];P=improfile(Ag,x,y);
for j=1:length(P);B(i,j)=P(j)';end
%u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
            C1=j;%dont break so get second line (lower)
        end;end;end
linedata=[C0 C1]
%-----
%remove the base line
n1=ca(2)-ca(1);nn=ca(1);
for i=1:n1
x=[nn+i nn+i];y=[290 535];P=improfile(Ag,x,y);
for qq=1:C0 P(qq)=0;end
P(C1-1)=(P(C1-2)+P(C1+1))/2;P(C1)=(P(C1-2)+P(C1+1))/2;
```

```

for s=1:length(P) %removes bright letters
    if P(s)==255;P(s)=0;end;end
for j=1:length(P)
    B(i,j)=P(j);end
%u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
%xlabel('scan down from top line');ylabel('pixel value - brightness')
%*****
%remove high noise
for j=1:length(P)-10%sums over 10 - forward
    D=0;
    for k=1:10
        D=D+P(j-1+k);end;Doo(j)=D/10;
        if ((j<C1)&(D/10>=50));G=j;break
        elseif j==C1;G=C1;break;end;end%use normally C1+3
for s=1:G
    P(s)=0;end
%*****
%sums over 10 - backwards
for j=1:length(P)
    q=length(P)+1-j;D=0;
    for k=1:10
        D=D+P(q+1-k);end;D2(i)=D;
        if ((q>C1)&(D/10>=50));G1=q;break%threshold usually 80
        elseif q==C1;G1=C1;break;end;end%use normally C1-3
for s=1:length(P)-G1
    q=length(P)+1-s;P(q)=0;end
%w=linspace(1,length(P),length(P));figure;plot(w,P);grid on
%*****
%smoother
F(i)=C1-G;
F0(i)=G;
F1(i)=C1-G1;
F00(i)=G1;
end
for k=1:50
for j=1:length(F)
    if ((j>=2)&(j<=length(F)-1)&(abs(F(j)-(F(j-1)+F(j+1))/2)>51-k))
        F(j)=(F(j-1)+F(j+1))/2;
    end;end;end
for k=1:50
for j=1:length(F1)
    if ((j>=2)&(j<=length(F1)-1)&(abs(F1(j)-(F1(j-1)+F1(j+1))/2)>51-k))
        F1(j)=(F1(j-1)+F1(j+1))/2;
    end;end;end
v=linspace(1,length(F),length(F));figure;plot(v,F,v,F1);grid on
%vv=linspace(1,length(F),length(F));figure;plot(vv,F+F1);grid on
%previous line adds data sets
%-----
%join two data sets via absolute value
for j=1:length(v)

```

```

    if F(j)>=-F1(j);FF(j)=F(j);
    else FF(j)=F1(j);end;end
%vvv=linspace(1,length(FF),length(FF));figure;plot(vvv,FF);grid on
%-----
%least squares fourier data fit.
y=FF;x=linspace(0.001,pi,length(y));
N = length(y); M = 50; % number of parameters
clear A;
for i = 1:N
    for j = 1:M
        A(i,j) =sin(j*x(i));
    end;end
clear b
for i=1:N
    b(i)=y(i);
end
% calculate parameters, a
a = (A'*A)^(-1) * (A'*b);
% calculate variances and covariances
C = (A'*A)^(-1);
% Draw points and curve fits.
z=0;
for i=1:M
    z=z+a(i,1)*sin(i*x);
end
figure;plot(v,z,v,FF);grid on
xlabel('distance from first click');ylabel('distance from v=0 line')
%-----
%find max and base of systole
for job = 1:length(z)
    if z(job)==max(z);job1=job;break;end;end;
for job2 = 1:length(z)
    if z(job1+job2)<0;job3=job2;break;end;end;fin=job3+job1-1;
for job4=1:length(z)
    if z(fin+job4)>0;job5=job4;break;end;end;fin2=fin+job4-1;
%-----
%area and ratio for above base line
%D3=pi*(ca(3)-ca(1))/(ca(2)-ca(1));Finds beginning of diastole by third click
%D3=input('enter D3:');%manual input
D3=pi*fin/(ca(2)-ca(1));Finds crossover point automatically
D4=pi*fin2/(ca(2)-ca(1));next cross over point
z1=0;
dx1=D3/1000;
for j=1:1000
    for i=1:M
        z1=z1+dx1*a(i,1)*sin(i*j*dx1);
    end;end
z2=0;
dx2=(D4-D3)/1000;
for j=1:1000

```

```

for i=1:M
    z2=z2+dx2*a(i,1)*sin(i*(D3+j*dx2));
end;end
z3=0;
dx3=(pi-D4)/1000;
for j=1:1000
    for i=1:M
        z3=z3+dx3*a(i,1)*sin(i*(D4+j*dx3));
    end;end
%-----
%total pixal value
%dopixcheck=input('enter dopixcheck:');
%if dopixcheck==1;n10=1;
%else n10=n1;end
for i=1:n1
    x=[nn+i nn+i];y=[290 558];P=improfile(Ag,x,y);
    for qq=1:C0 P(qq)=0;end
    P(C1-1)=(P(C1-2)+P(C1+1))/2;P(C1)=(P(C1-2)+P(C1+1))/2;
    for s=1:length(P) %removes bright letters
        if P(s)==255;P(s)=0;end;end
    E=0;
    for j=1:length(P)
        E=E+P(j);end
    pv(i)=E;
    E1=0;pp1=0;
    for j=1:C1-F(i)%F0(i)
        E1=E1+P(j);pp1=pp1+1;end
    pvenvF0(i)=E1;pnum1(i)=pp1;
    E2=0;pp2=0;
    for j=1:C1-F1(i)%F00(i)
        E2=E2+P(j);pp2=pp2+1;end
    pvenvF00(i)=E2;pnum2(i)=pp2;
end
PVtot=sum(pv);
PVenvF0=sum(pvenvF0);
PVenvF00=sum(pvenvF00);
PVimage=(PVenvF00-PVenvF0);
PVratio=PVimage/PVtot;
numave=sum(pnum2)-sum(pnum1);
PVave=PVimage/numave;
sysmax=max(z);
sysmin=min(z);
pp=sysmax-sysmin;
mufreq=mean(z);
pindex=pp/mufreq;%useful for state of artery and impedance
%-----
qa1=0;
for i=1:length(a)
    qa1=qa1+a(i)^2;end;
qa2=0;

```

```

for j=1:ca(2)-(ca(1)+fin)
    qa2=qa2+z(fin+j-1);end;qa3=qa2/(ca(2)-(ca(1)+fin));
    Porig=qa1/qa3^2;
    datt=[z1 z2 z3 Porig]% Three separate areas and loupas

```

2 *Change to code for edge detection*

```

%edge detection
clear all
file=uigetfile('*.tif');
a=imread(file);
b1=rgb2gray(a);
[D1 D2]=size(b1);
b1(1:278,1:768)=0;
%figure;imshow(b1,'notruesize');pixval
%-----
%scan image to find j value of top line from first 5 scans
n=5;
for i=1:n
    x=[325+i 325+i];y=[290 520];P=improfile(b1,x,y);
    for j=1:length(P);B(i,j)=P(j)';end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C1=j;break
            end;end;end
%-----
%scan image to find j value of V=0 line from first 5 scans
n=5;
for i=1:n
    x=[325+i 325+i];y=[290 520];P=improfile(b1,x,y);
    for j=1:length(P);B(i,j)=P(j)';end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C2=j;%dont break so get second line (lower)
            end;end;end
%-----
m=20;
for i=1:D1
    for j=1:D2

```

```

    A(i,j)=exp(-((i-D1/2)^2+(j-D2/2)^2)/(2*m^2));
    if b1(i,j)==255;b1(i,j)=0;end
end;end
%figure;imshow(b1,'notruesize');
c=real(ifft2(ifftshift((fftshift(fft2(b1))).*A/255)));
d=edge(c,'sobel',(graythresh(c) * 0.1),'both');
figure;imshow(d,'notruesize');[ca r p]=impixel;
%-----
%scan profile down
n1=ca(2)-ca(1);nn=ca(1);%input('enter n1:');nn=input('enter nn:');
for i=1:n1
x=[nn+i nn+i];y=[300 520];P=improfile(d,x,y);
for qq=1:C1 P(qq)=0;end
for j=1:length(P)
    if P(j)==1;C3(i)=C2-9-j;break;%-9 calibrated on a few images
    end;end
%*****
%smoother down
F=C3;
for k=1:50
for j=1:length(F)
    if ((j>=2)&(j<=length(F)-1)&(abs(F(j)-(F(j-1)+F(j+1))/2)>51-k))
        F(j)=(F(j-1)+F(j+1))/2;
    end;end;end
%-----
%scan profile up
for j=1:length(P)
    q=length(P)+1-j;
    if P(q)==1;C4(i)=C2-9-q;break
    end;end;end
%smoother up
F1=C4;
for k=1:50
for j=1:length(F1)
    if ((j>=2)&(j<=length(F1)-1)&(abs(F1(j)-(F1(j-1)+F1(j+1))/2)>51-k))
        F1(j)=(F1(j-1)+F1(j+1))/2;
    end;end;end
u=linspace(0,length(F),length(F));figure;plot(u,C3,u,C4);grid on
xlabel('distance from first click');ylabel('distance from v=0 line')
%-----
%decides which is bigger(absolute value)
for j=1:length(F)
    if F(j)>=F1(j);FF(j)=F(j);
    else FF(j)=F1(j);end;end
v=linspace(1,length(FF),length(FF));figure;plot(v,F,v,F1);grid on
xlabel('distance from first click');ylabel('distance from v=0 line')
%-----
%least squares fourier data fit.
y=FF;x=linspace(0.001,pi,length(y));
N = length(y); M = 50; % number of parameters

```

```

clear A;
for i = 1:N
    for j = 1:M
        A(i,j) = sin(j*x(i));
    end
end
clear b
for i=1:N
    b(i)=y(i);
end
% calculate parameters, a
a = (A'*A)^(-1) * (A'*b);
% calculate variances and covariances
C = (A'*A)^(-1);
% Draw points and curve fits.
z=0;
for i=1:M
    z=z+a(i,1)*sin(i*x);
end
figure;plot(v,z,v,FF);grid on
xlabel('distance from first click');ylabel('distance from v=0 line')
%-----
%area and ratio for above base line
%D3=input('enter D3:');%manual input
D3=pi*(ca(3)-ca(1))/(ca(2)-ca(1));%input('enter D3:');
z1=0;
dx1=D3/1000;
for j=1:1000
    for i=1:M
        z1=z1+dx1*a(i,1)*sin(i*j*dx1);
    end;end
z1;
z2=0;
for j=1:1000
    dx2=(pi-D3)/1000;
    for i=1:M
        z2=z2+dx2*a(i,1)*sin(i*(D3+j*dx2));
    end;end
z2;
ratio=z1/z2;
%-----
%average pixel value
for i=1:n1
    x=[nn+i nn+i];y=[290 532];P=improfile(b1,x,y);
    for qq=1:C1 P(qq)=0;end
    P(C2-1)=(P(C2-2)+P(C2+1))/2;P(C2)=(P(C2-2)+P(C2+1))/2;
    for s=1:length(P) %removes bright letters
        if P(s)==255;P(s)=0;end;end
    E=0;
    for j=1:length(P)

```

```

    E=E+P(j);end
pv(i)=E;
E1=0;pp1=0;
for j=1:C2-F(i)
    E1=E1+P(j);pp1=pp1+1;end
pvenvF(i)=E1;pnum1(i)=pp1;
E2=0;pp2=0;
for j=1:C2-F1(i)
    E2=E2+P(j);pp2=pp2+1;end
pvenvF1(i)=E2;pnum2(i)=pp2;
end
PVtot=sum(pv);
PVenvF=sum(pvenvF);
PVenvF1=sum(pvenvF1);
PVimage=(PVenvF1-PVenvF);
PVratio=PVimage/PVtot;
numave=sum(pnum2)-sum(pnum1);
PVave=PVimage/numave;
%-----
qa1=0;%loupas(PIorig)
for i=1:length(a)
    qa1=qa1+a(i)^2;end;
qa2=0;
    for j=1:ca(2)-ca(3)
        qa2=qa2+z(ca(3)-ca(1)+j-1);end;qa3=qa2/(ca(2)-ca(3));
    PIorig=qa1/qa3^2;
datt=[ratio PIorig PVave]

```

3. Code change for differential method

```

%differential
clear all
%-----
%locate and display image
file=uigetfile('*.tif');
A=imread(file);
Ag=rgb2gray(A);%Ag=imhmin(Ag,100);
Ag(1:278,1:768)=0;
figure;imshow(Ag,'notruesize');[ca r p]=impixel;
%-----
%scan image to find j value of top line from first 5 scans
n=5;
for k=1:n
    x=[325+k 325+k];y=[290 558];P=improfile(Ag,x,y);
    for j=1:length(P);B(k,j)=P(j)';end

```



```

%u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C0=j;break%break so get top line
            end;end;end
%-----
%scan image to find j value of V=0 line from first 5 scans
n=5;
for i=1:n
    x=[325+i 325+i]; %avoid writing
    y=[290 558];
    P=improfile(Ag,x,y);
    for j=1:length(P)
        B(i,j)=P(j)';
    end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C1=j;%dont break so get second line (lower)
            end;end;end
C1
%-----
%remove the base line
%Ag=medfilt2(Ag,[3 3]);
%remove the base line
n1=ca(2)-ca(1);nn=ca(1);
for k=1:n1
    x=[nn+k nn+k];y=[290 558];P=improfile(Ag,x,y);
    for qq=1:C0 P(qq)=0;end
    P(C1-1)=(P(C1-2)+P(C1+1))/2;
    P(C1)=(P(C1-2)+P(C1+1))/2;
    for s=1:length(P) %removes bright letters
        if P(s)==255
            P(s)=0;
        end;end
    for j=1:length(P)
        B(k,j)=P(j)';
    end
    %u=linspace(0,length(B),length(B));
    %figure;plot(u,B(k,:));grid on
    %*****same run as n=200
    %least squares
    % fourier data fit.
    y=P;

```

```

x=linspace(0.001,pi,length(y));
N = length(y);
M = 100; % number of parameters
clear A;
for i = 1:N
    for j = 1:M
        A(i,j) =sin(j*x(i));
    end;end
clear b
for i=1:N
    b(i)=y(i);end
% calculate parameters, a
a = (A'*A)^(-1) * (A'*b');
% calculate variances and covariances
C = (A'*A)^(-1);
A;
% Draw points and curve fits.
z=0;
for i=1:M
    z=z+a(i,1)*sin(i*x);
end
%first differential
h=0.00001;
z1=0;
z2=0;
M=100;
for i=1:M
    z1=z1+a(i,1)*sin(i*(x+h));
end
for i=1:M
    z2=z2+a(i,1)*sin(i*x);
end
dzdx=(z2-z1)/h;
%figure;plot(x,y,x,z);grid on
%figure;plot(x,dzdx);grid on
for j=1:length(dzdx)
    if ((abs(dzdx(j))>=1500)&(j<=C1-1))
        z3(k)=j;break
    elseif j>C1-1
        z3(k)=C1-1;
    end;end
%*****
for j=1:length(dzdx)
    s=length(dzdx)+1-j;
    if ((abs(dzdx(s))>=1500)&(s>=C1-1))
        w3(k)=s;
        break
    elseif s<C1-1
        w3(k)=C1-1;
    end;end

```

```

end
z4=C1-1-z3;
z5=linspace(1,length(z4),length(z4));
%figure;plot(z5,z4);grid on
w4=C1-1-w3;
w5=linspace(1,length(w4),length(w4));
%figure;plot(w5,w4,z5,z4);grid on
%figure;plot(w5,w4+z4);grid on
%-----
for k=1:50
for i=1:length(z4)
    if ((i>=2)&(i<=length(z4)-1)&(abs(z4(i)-(z4(i-1)+z4(i+1))/2)>51-k))
        z4(i)=(z4(i-1)+z4(i+1))/2;
    end;end;end
%figure;plot(z5,z4);grid on
%*****
for k=1:50
for i=1:length(w4)
    if ((i>=2)&(i<=length(w4)-1)&(abs(w4(i)-(w4(i-1)+w4(i+1))/2)>51-k))
        w4(i)=(w4(i-1)+w4(i+1))/2;
    end;end;end
figure;plot(z5,z4,w5,w4);grid on
xlabel('distance from first click');ylabel('distance from v=0 line')
%figure;plot(z5,z4+w4);grid on
%-----
%join two data sets via absolute value
for j=1:length(z4)
    if z4(j)>=abs(w4(j));FF(j)=z4(j);
    else FF(j)=w4(j);end;end
%-----
w=z4+w4;
%least squares
% fourier data fit.
y=z4;
x=linspace(0.001,pi,length(y));
N = length(y);
M = 50; % number of parameters
clear A;
for i = 1:N
    for j = 1:M
        A(i,j) =sin(j*x(i));
    end
end
clear b
for i=1:N
    b(i)=y(i);
end
% calculate parameters, a
a = (A'*A)^(-1) * (A'*b);
% calculatate variances and covariances

```

```

C = (A'*A)^(-1);
A;
% Calculate chi-square.
%t=0;
%for i=1:N
%   t=t+(y(i)-a(1,1)*sin(x(i))-a(2,1)*sin(2*x(i))-a(3,1)*sin(3*x(i)));
%end
%chisqr=t;
% Draw points and curve fits.
z=0;
for i=1:M
    z=z+a(i,1)*sin(i*x);
end
figure;plot(x,z,x,FF);grid on
%-----
%area and ratio for above base line
D3=pi*(ca(3)-ca(1))/(ca(2)-ca(1));
%D3=input('enter D3:');%manual input
z1=0;
dx1=D3/1000;
for j=1:1000
    for i=1:M
        z1=z1+dx1*a(i,1)*sin(i*j*dx1);
    end;end
z2=0;
for j=1:1000
    dx2=(pi-D3)/1000;
    for i=1:M
        z2=z2+dx2*a(i,1)*sin(i*(D3+j*dx2));
    end;end
ratio=(z1/z2);
end
%figure;imshow(Ag,'notruesize');paxval
%-----
%total pixal value
%dopixcheck=input('enter dopixcheck:');
%if dopixcheck==1;n10=1;
%else n10=n1;end
for i=1:n1
    x=[nn+i nn+i];y=[290 558];P=improfile(Ag,x,y);
    for qq=1:C0 P(qq)=0;end
    P(C1-1)=(P(C1-2)+P(C1+1))/2;P(C1)=(P(C1-2)+P(C1+1))/2;
    for s=1:length(P) %removes bright letters
        if P(s)==255;P(s)=0;end;end
    E=0;
    for j=1:length(P)
        E=E+P(j);end
    pv(i)=E;
    E1=0;pp1=0;
    for j=1:C1-1-z4(i)

```

```

    E1=E1+P(j);pp1=pp1+1;end
pvenvtop(i)=E1;pnum1(i)=pp1;
E2=0;pp2=0;
for j=1:C1-1-w4(i)
    E2=E2+P(j);pp2=pp2+1;end
pvenvbot(i)=E2;pnum2(i)=pp2;
end
PVtot=sum(pv)
PVenvtop=sum(pvenvtop);
PVenvbot=sum(pvenvbot);
PVimage=(PVenvbot-PVenvtop)
PVratio=PVimage/PVtot
numave=sum(pnum2)-sum(pnum1);
PVave=PVimage/numave
sysmax=max(FF);
sysmin=min(FF);
for i=1:length(FF)
    if FF(i)==min(FF);gg=i;end;end
for i=1:length(FF)
    if i<=gg;FF(i)=0;end;end
dysmax=max(FF);
heightdata=[sysmax sysmin dysmax]
ratio

```

4. Code change for percentile method

```

%percentile
clear all
%-----
%locate and display image
file=uigetfile('*.tif');
A=imread(file);
Ag=rgb2gray(A);%Ag=imhmin(Ag,100);
Ag(1:278,1:768)=0;
figure;imshow(Ag,'notruesize');[ca r p]=impixel;
%-----
%scan image to find j value of top line from first 5 scans
n=5;
for k=1:n
    x=[325+k 325+k];y=[290 558];P=improfile(Ag,x,y);
    for j=1:length(P);B(k,j)=P(j);end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n

```

```

    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C0=j;break;%break so get top line
            end;end;end
%-----
%scan image to find j value of V=0 line from first 5 scans
n=5;
for k=1:n
    x=[325+k 325+k];y=[290 558];P=improfile(Ag,x,y);
    for j=1:length(P);B(k,j)=P(j)';end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
end
for i=1:n
    for j=1:length(B)
        if
            ((B(1,j)==255)&(B(2,j)==255)&(B(3,j)==255)&(B(4,j)==255)&(B(5,j)==255))
                C1=j;%dont break so get second line (lower)
            end;end;end
%-----
%Ag=medfilt2(Ag,[3 3]);
%remove the base line
n1=ca(2)-ca(1);nn=ca(1);
for i=1:n1
    x=[nn+i nn+i];y=[290 558];P=improfile(Ag,x,y);
    for qq=1:C0+4 P(qq)=0;end
    P(C1-1)=(P(C1-2)+P(C1+1))/2;P(C1)=(P(C1-2)+P(C1+1))/2;
    for s=1:length(P) %removes bright letters
        if P(s)==255;P(s)=0;end;end
    for j=1:length(P)
        B(i,j)=P(j)';end
    %u=linspace(0,length(B),length(B));figure;plot(u,B(i,:));grid on
    %*****
    %check out total power for each spectral output - forward
    D=0;
    for m=1:length(P)
        D=D+P(m);end;D1(i)=D;
    E=0;
    for mm=1:length(P)
        E=E+P(mm);
        if E>0.50*D; h(i)=C1-mm;break;D2(i)=E;
        end;end
    %end
    for k=1:50
        for j=1:length(h)
            if ((j>=2)&(j<=length(h)-1)&(abs(h(j)-(h(j-1)+h(j+1))/2)>51-k))
                h(j)=(h(j-1)+h(j+1))/2;
            end;end;end
        %*****
        %check out total power - reverse

```

```

D1=0;
for mm1=1:length(P)
    q=length(P)+1-mm1;
    D1=D1+P(q);end
    E1=0;
    for mm2=1:length(P)
        q=length(P)+1-mm2;
        E1=E1+P(q);
        if E1>0.50*D1;h1(i)=C1-q;break
        end;end
    for k=1:50
        for j=1:length(h1)
            if ((j>=2)&(j<=length(h1)-1)&(abs(h1(j)-(h1(j-1)+h1(j+1))/2)>51-k))
                h1(j)=(h1(j-1)+h1(j+1))/2;
            end;end;end
        end
        v=linspace(0,length(h),length(h));figure;plot(v,h,v,h1);grid on
        %join two data sets via importance
        for j=1:length(h)
            if h(j)>=abs(h1(j));FF(j)=h(j);
            else FF(j)=h1(j);end;end
        %-----
        %least squares fourier data fit.
        y=FF;x=linspace(0.001,pi,length(y));
        N = length(y); M = 50; % number of parameters
        clear A;
        for i = 1:N
            for j = 1:M
                A(i,j) =sin(j*x(i));
            end
        end
        clear b
        for i=1:N
            b(i)=y(i);
        end
        % calculate parameters, a
        a = (A'*A)^(-1) * (A'*b');
        % calculate variances and covariances
        C = (A'*A)^(-1);
        % Draw points and curve fits.
        z=0;
        for i=1:M
            z=z+a(i,1)*sin(i*x);
        end
        figure;plot(v,z,v,FF);grid on
        xlabel('distance from first click');ylabel('distance from v=0 line')
        %-----
        %area and ratio for above base line
        D3=pi*(ca(3)-ca(1))/(ca(2)-ca(1));%input('enter D3:');
        z1=0;

```

```

dx1=D3/1000;
for j=1:1000
for i=1:M
    z1=z1+dx1*a(i,1)*sin(i*j*dx1);
end;end
z2=0;
for j=1:1000
dx2=(pi-D3)/1000;
for i=1:M
    z2=z2+dx2*a(i,1)*sin(i*(D3+j*dx2));
end;end
data=[ z1 z2 ];
ratio=z1/z2
%-----
%total pixal value
%dopixcheck=input('enter dopixcheck:');
%if dopixcheck==1;n10=1;
%else n10=n1;end
for i=1:n1
x=[nn+i nn+i];y=[290 558];P=improfile(Ag,x,y);
for qq=1:C0 P(qq)=0;end
P(C1-1)=(P(C1-2)+P(C1+1))/2;P(C1)=(P(C1-2)+P(C1+1))/2;
for s=1:length(P) %removes bright letters
    if P(s)==255;P(s)=0;end;end
E=0;
for j=1:length(P)
    E=E+P(j);end
pv(i)=E;
E1=0;pp1=0;
for j=1:C1-h(i)
    E1=E1+P(j);pp1=pp1+1;end
pvenvtop(i)=E1;pnum1(i)=pp1;
E2=0;pp2=0;
for j=1:C1-h1(i)
    E2=E2+P(j);pp2=pp2+1;end
pvenvbot(i)=E2;pnum2(i)=pp2;
end
PVtot=sum(pv);
PVenenvtop=sum(pvenvtop);
PVenenvbot=sum(pvenvbot);
PVimage=(PVenenvbot-PVenenvtop);
PVratio=PVimage/PVtot;
numave=sum(pnum2)-sum(pnum1);
PVave=PVimage/numave

```